
MODULE *HPaxos*

EXTENDS *Integers, TLAPS, TLC*

$Ballot \triangleq Nat$

LEMMA $BallotLeqTrans \triangleq$

ASSUME NEW $A \in Ballot$, NEW $B \in Ballot$, NEW $C \in Ballot$, $A \leq B$, $B \leq C$ PROVE $A \leq C$
 PROOF BY DEF $Ballot$

LEMMA $BallotLeLeqTrans \triangleq$

ASSUME NEW $A \in Ballot$, NEW $B \in Ballot$, NEW $C \in Ballot$, $A < B$, $B \leq C$ PROVE $A < C$
 PROOF BY DEF $Ballot$

LEMMA $BallotLeqLeTrans \triangleq$

ASSUME NEW $A \in Ballot$, NEW $B \in Ballot$, NEW $C \in Ballot$, $A \leq B$, $B < C$ PROVE $A < C$
 PROOF BY DEF $Ballot$

LEMMA $BallotLeNotLeq \triangleq$ ASSUME NEW $A \in Ballot$, NEW $B \in Ballot$, $A < B$ PROVE $\neg B \leq A$

PROOF BY DEF $Ballot$

LEMMA $BallotOrderCases \triangleq$ ASSUME NEW $A \in Ballot$, NEW $B \in Ballot$ PROVE $A < B \vee B < A \vee A = B$

PROOF BY DEF $Ballot$

CONSTANT $Value$

ASSUME $ValueNotEmpty \triangleq Value \neq \{\}$

$None \triangleq \text{CHOOSE } v : v \notin Value$

CONSTANTS $Acceptor$,
 $SafeAcceptor$,
 $FakeAcceptor$,
 $ByzQuorum$,
 $Learner$

ASSUME $SafeAcceptorAssumption \triangleq$

$\wedge SafeAcceptor \cap FakeAcceptor = \{\}$
 $\wedge SafeAcceptor \cup FakeAcceptor = Acceptor$

LEMMA $SafeAcceptorIsAcceptor \triangleq SafeAcceptor \subseteq Acceptor$

PROOF BY $SafeAcceptorAssumption$

LEMMA $FakeAcceptorIsAcceptor \triangleq FakeAcceptor \subseteq Acceptor$

PROOF BY $SafeAcceptorAssumption$

ASSUME $BQAssumption \triangleq$

$\wedge \forall Q \in ByzQuorum : Q \subseteq Acceptor$

ASSUME *BallotAssumption* \triangleq
 $\wedge (Ballot \cup \{-1\}) \cap Acceptor = \{\}$
 $\wedge (Ballot \cup \{-1\}) \cap ByzQuorum = \{\}$
 $\wedge (Ballot \cup \{-1\}) \cap Learner = \{\}$

Learner graph

CONSTANT *TrustLive*
ASSUME *TrustLiveAssumption* \triangleq *TrustLive* \in SUBSET [*lr* : *Learner*, *q* : *ByzQuorum*]

CONSTANT *TrustSafe*
ASSUME *TrustSafeAssumption* \triangleq *TrustSafe* \in SUBSET [*from* : *Learner*, *to* : *Learner*, *q* : *ByzQuorum*]

ASSUME *LearnerGraphAssumption* \triangleq
symmetry
 $\wedge \forall E \in TrustSafe :$
 $[from \mapsto E.to, to \mapsto E.from, q \mapsto E.q] \in TrustSafe$
transitivity
 $\wedge \forall E1, E2 \in TrustSafe :$
 $E1.q = E2.q \wedge E1.to = E2.from \Rightarrow$
 $[from \mapsto E1.from, to \mapsto E2.to, q \mapsto E1.q] \in TrustSafe$
closure
 $\wedge \forall E \in TrustSafe : \forall Q \in ByzQuorum :$
 $E.q \subseteq Q \Rightarrow$
 $[from \mapsto E.from, to \mapsto E.to, q \mapsto Q] \in TrustSafe$
validity
 $\wedge \forall E \in TrustSafe : \forall Q1, Q2 \in ByzQuorum :$
 $[lr \mapsto E.from, q \mapsto Q1] \in TrustLive \wedge$
 $[lr \mapsto E.to, q \mapsto Q2] \in TrustLive \Rightarrow$
 $\exists N \in E.q : N \in Q1 \wedge N \in Q2$

CONSTANT *Ent*
ASSUME *EntanglementAssumption* \triangleq
 $\wedge Ent \in SUBSET (Learner \times Learner)$
 $\wedge \forall L1, L2 \in Learner :$
 $\langle L1, L2 \rangle \in Ent \equiv$
 $[from \mapsto L1, to \mapsto L2, q \mapsto SafeAcceptor] \in TrustSafe$

LEMMA *EntanglementSym* \triangleq
ASSUME NEW *L1* \in *Learner*, NEW *L2* \in *Learner*, $\langle L1, L2 \rangle \in Ent$ PROVE $\langle L2, L1 \rangle \in Ent$
PROOF BY *EntanglementAssumption*, *LearnerGraphAssumption*

LEMMA *EntanglementSelf* \triangleq
ASSUME NEW *L1* \in *Learner*, NEW *L2* \in *Learner*, $\langle L1, L2 \rangle \in Ent$ PROVE $\langle L1, L1 \rangle \in Ent$
PROOF BY *EntanglementAssumption*, *LearnerGraphAssumption*

LEMMA *EntanglementTrustLive* \triangleq
 ASSUME NEW $L1 \in \text{Learner}$, NEW $L2 \in \text{Learner}$,
 NEW $Q1 \in \text{ByzQuorum}$, NEW $Q2 \in \text{ByzQuorum}$,
 $\langle L1, L2 \rangle \in \text{Ent}$,
 $[lr \mapsto L1, q \mapsto Q1] \in \text{TrustLive}$,
 $[lr \mapsto L2, q \mapsto Q2] \in \text{TrustLive}$
 PROVE $\exists N \in \text{SafeAcceptor} : N \in Q1 \wedge N \in Q2$
 PROOF BY *EntanglementAssumption*, *LearnerGraphAssumption*

Messages

Message \triangleq
 $[type : \{ \text{"1a"} \}, lr : \text{Learner}, bal : \text{Ballot}] \cup$
 $[$
 $type : \{ \text{"1b"} \},$
 $lr : \text{Learner},$
 $acc : \text{Acceptor},$
 $bal : \text{Ballot},$
 $votes : \text{SUBSET } [lr : \text{Learner}, bal : \text{Ballot}, val : \text{Value}],$
 $proposals : \text{SUBSET } [lr : \text{Learner}, bal : \text{Ballot}, val : \text{Value}]$
 $] \cup$
 $[type : \{ \text{"1c"} \}, lr : \text{Learner}, bal : \text{Ballot}, val : \text{Value}] \cup$
 $[type : \{ \text{"2av"} \}, lr : \text{Learner}, acc : \text{Acceptor}, bal : \text{Ballot}, val : \text{Value}] \cup$
 $[type : \{ \text{"2b"} \}, lr : \text{Learner}, acc : \text{Acceptor}, bal : \text{Ballot}, val : \text{Value}]$

Algorithm specification

VARIABLES $maxBal$,
 $votesSent$,
 $2avSent$,
 $received$,
 $connected$,
 $receivedByLearner$,
 $decision$,
 $msgs$

InitializedBallot(lr, bal) \triangleq
 $\exists m \in msgs : m.type = \text{"1a"} \wedge m.lr = lr \wedge m.bal = bal$

AnnouncedValue(lr, bal, val) \triangleq
 $\exists m \in msgs : m.type = \text{"1c"} \wedge m.bal = bal \wedge m.val = val$

ChosenIn(lr, bal, v) \triangleq
 $\exists Q \in \text{ByzQuorum} :$
 $\wedge [lr \mapsto lr, q \mapsto Q] \in \text{TrustLive}$

$$\begin{aligned}
& \wedge \forall aa \in Q : \\
& \quad \exists m \in \{mm \in receivedByLearner[lr] : mm.bal = bal\} : \\
& \quad \quad \wedge m.val = v \\
& \quad \quad \wedge m.acc = aa \\
\\
& KnowsSafeAt1(l, ac, b, v) \triangleq \\
& \quad LET \ S \triangleq \{mm \in received[ac] : mm.type = \text{"1b"} \wedge mm.lr = l \wedge mm.bal = b\} \\
& \quad IN \quad \exists BQ \in ByzQuorum : \\
& \quad \quad \wedge [lr \mapsto l, q \mapsto BQ] \in TrustLive \\
& \quad \quad \wedge \forall a \in BQ : \\
& \quad \quad \quad \exists m \in S : \\
& \quad \quad \quad \quad \wedge m.acc = a \\
& \quad \quad \quad \quad \wedge \forall p \in \{pp \in m.votes : \langle pp.lr, l \rangle \in connected[ac]\} : \\
& \quad \quad \quad \quad \quad b \leq p.bal \\
\\
& KnowsSafeAt2(l, ac, b, v) \triangleq \\
& \quad LET \ S \triangleq \{mm \in received[ac] : mm.type = \text{"1b"} \wedge mm.lr = l \wedge mm.bal = b\} \\
& \quad IN \quad \exists c \in Ballot : \\
& \quad \quad \wedge c < b \\
& \quad \quad \wedge \exists BQ \in ByzQuorum : \\
& \quad \quad \quad \wedge [lr \mapsto l, q \mapsto BQ] \in TrustLive \\
& \quad \quad \quad \wedge \forall a \in BQ : \\
& \quad \quad \quad \quad \exists m \in S : \\
& \quad \quad \quad \quad \quad \wedge m.acc = a \\
& \quad \quad \quad \quad \quad \wedge \forall p \in \{pp \in m.votes : \langle pp.lr, l \rangle \in connected[ac]\} : \\
& \quad \quad \quad \quad \quad \quad \wedge p.bal \leq c \\
& \quad \quad \quad \quad \quad \quad \wedge (p.bal = c) \Rightarrow (p.val = v) \\
& \quad \quad \wedge \exists WQ \in ByzQuorum : \\
& \quad \quad \quad \wedge [lr \mapsto l, q \mapsto WQ] \in TrustLive \\
& \quad \quad \quad \wedge \forall a \in WQ : \\
& \quad \quad \quad \quad \exists m \in S : \\
& \quad \quad \quad \quad \quad \wedge m.acc = a \\
& \quad \quad \quad \quad \quad \wedge \exists p \in m.proposals : \\
& \quad \quad \quad \quad \quad \quad \wedge p.lr = l \\
& \quad \quad \quad \quad \quad \quad \wedge p.bal = c \\
& \quad \quad \quad \quad \quad \quad \wedge p.val = v \\
\\
& KnowsSafeAt(l, ac, b, v) \triangleq \\
& \quad \vee KnowsSafeAt1(l, ac, b, v) \\
& \quad \vee KnowsSafeAt2(l, ac, b, v) \\
\\
& vars \triangleq \langle maxBal, votesSent, 2avSent, received, connected, receivedByLearner, decision, msgs \rangle \\
\\
& TypeOK \triangleq \\
& \quad \wedge \quad msgs \in SUBSET \ Message \\
& \quad \wedge \quad maxBal \in [Learner \times Acceptor \rightarrow Ballot]
\end{aligned}$$

$$\begin{aligned}
& \wedge \text{ votesSent} \in [\text{Acceptor} \rightarrow \text{SUBSET } [lr : \text{Learner}, bal : \text{Ballot}, val : \text{Value}]] \\
& \wedge \text{ 2avSent} \in [\text{Acceptor} \rightarrow \text{SUBSET } [lr : \text{Learner}, bal : \text{Ballot}, val : \text{Value}]] \\
& \wedge \text{ connected} \in [\text{Acceptor} \rightarrow \text{SUBSET } (\text{Learner} \times \text{Learner})] \\
& \wedge \text{ received} \in [\text{Acceptor} \rightarrow \text{SUBSET } \text{Message}] \\
& \wedge \text{ receivedByLearner} \in [\text{Learner} \rightarrow \text{SUBSET } \text{Message}] \\
& \wedge \text{ decision} \in [\text{Learner} \times \text{Ballot} \rightarrow \text{SUBSET } \text{Value}]
\end{aligned}$$

$$\begin{aligned}
\text{Init} & \triangleq \\
& \wedge \text{ msgs} = \{\} \\
& \wedge \forall L \in \text{Learner} : \forall A \in \text{SafeAcceptor} : \text{maxBal}[L, A] = 0 \\
& \wedge \forall A \in \text{SafeAcceptor} : \text{2avSent}[A] = \{\} \\
& \wedge \forall A \in \text{SafeAcceptor} : \text{votesSent}[A] = \{\} \\
& \wedge \forall A \in \text{SafeAcceptor} : \text{connected}[A] = \text{Learner} \times \text{Learner} \\
& \wedge \forall A \in \text{Acceptor} : \text{received}[A] = \{\} \\
& \wedge \forall L \in \text{Learner} : \text{receivedByLearner}[L] = \{\} \\
& \wedge \forall L \in \text{Learner} : \forall B \in \text{Ballot} : \text{decision}[L, B] = \{\} \\
& \wedge \text{TypeOK}
\end{aligned}$$

$$\text{Send}(m) \triangleq \text{msgs}' = \text{msgs} \cup \{m\}$$

$$\begin{aligned}
\text{Phase1a}(l, b) & \triangleq \\
& \wedge \text{ Send}([type \mapsto \text{"1a"}, lr \mapsto l, bal \mapsto b]) \\
& \wedge \text{ UNCHANGED } \langle \text{maxBal}, \text{votesSent}, \text{2avSent}, \text{received}, \text{connected}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{Phase1c}(l, b, v) & \triangleq \\
& \wedge \text{ Send}([type \mapsto \text{"1c"}, lr \mapsto l, bal \mapsto b, val \mapsto v]) \\
& \wedge \text{ UNCHANGED } \langle \text{maxBal}, \text{votesSent}, \text{2avSent}, \text{received}, \text{connected}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{MaxVote}(a, b, \text{vote}) & \triangleq \\
& \wedge \text{ vote.bal} < b \\
& \wedge \forall \text{ other} \in \text{votesSent}[a] : \\
& \quad \text{other.lr} = \text{vote.lr} \wedge \text{other.bal} < b \Rightarrow \\
& \quad \text{other.bal} \leq \text{vote.bal}
\end{aligned}$$

$$\begin{aligned}
\text{Phase1b}(l, b, a) & \triangleq \\
& \wedge \text{ maxBal}[l, a] \leq b \\
& \wedge \text{ InitializedBallot}(l, b) \\
& \wedge \text{ maxBal}' = [\text{maxBal} \text{ EXCEPT } ![l, a] = b] \\
& \wedge \text{ Send}([\\
& \quad type \mapsto \text{"1b"}, \\
& \quad lr \mapsto l, \\
& \quad acc \mapsto a, \\
& \quad bal \mapsto b, \\
& \quad \text{votes} \mapsto \{p \in \text{votesSent}[a] : \text{MaxVote}(a, b, p)\}, \\
& \quad \text{proposals} \mapsto \{p \in \text{2avSent}[a] : p.bal < b \wedge p.lr = l\} \\
& \quad \left. \right]) \\
& \wedge \text{ UNCHANGED } \langle \text{votesSent}, \text{2avSent}, \text{received}, \text{connected}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{Phase2av}(l, b, a, v) &\triangleq \\
&\wedge \text{maxBal}[l, a] \leq b \\
&\wedge \text{InitializedBallot}(l, b) \\
&\wedge \text{AnnouncedValue}(l, b, v) \\
&\wedge \forall P \in \{p \in 2\text{avSent}[a] : p.\text{bal} = b \wedge \langle p.\text{lr}, l \rangle \in \text{connected}[a]\} : P.\text{val} = v \\
&\wedge \text{KnowsSafeAt}(l, a, b, v) \\
&\wedge \text{Send}([type \mapsto \text{"2av"}, lr \mapsto l, acc \mapsto a, bal \mapsto b, val \mapsto v]) \\
&\wedge 2\text{avSent}' = [2\text{avSent} \text{ EXCEPT } ![a] = 2\text{avSent}[a] \cup \{[lr \mapsto l, bal \mapsto b, val \mapsto v]\}] \\
&\wedge \text{UNCHANGED} \langle \text{maxBal}, \text{votesSent}, \text{received}, \text{connected}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{Phase2b}(l, b, a, v) &\triangleq \\
&\wedge \forall L \in \text{Learner} : \text{maxBal}[L, a] \leq b \\
&\wedge \exists Q \in \text{ByzQuorum} : \\
&\quad \wedge [lr \mapsto l, q \mapsto Q] \in \text{TrustLive} \\
&\quad \wedge \forall aa \in Q : \\
&\quad \quad \exists m \in \{mm \in \text{received}[a] : \\
&\quad \quad \quad \wedge mm.\text{type} = \text{"2av"} \\
&\quad \quad \quad \wedge mm.\text{lr} = l \\
&\quad \quad \quad \wedge mm.\text{bal} = b\} : \\
&\quad \quad \wedge m.\text{val} = v \\
&\quad \quad \wedge m.\text{acc} = aa \\
&\wedge \text{Send}([type \mapsto \text{"2b"}, lr \mapsto l, acc \mapsto a, bal \mapsto b, val \mapsto v]) \\
&\wedge \text{votesSent}' = [\text{votesSent} \text{ EXCEPT } ![a] = \\
&\quad \text{votesSent}[a] \cup \{[lr \mapsto l, bal \mapsto b, val \mapsto v]\}] \\
&\wedge \text{UNCHANGED} \langle \text{maxBal}, 2\text{avSent}, \text{received}, \text{connected}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{Recv}(l, a) &\triangleq \\
&\wedge \exists m \in \text{msgs} : \text{received}' = [\text{received} \text{ EXCEPT } ![a] = \text{received}[a] \cup \{m\}] \\
&\wedge \text{UNCHANGED} \langle \text{msgs}, \text{maxBal}, 2\text{avSent}, \text{votesSent}, \text{connected}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{Disconnect}(a) &\triangleq \\
&\wedge \exists P \in \text{SUBSET} \{LL \in \text{Learner} \times \text{Learner} : LL \notin \text{Ent}\} : \\
&\quad \text{connected}' = [\text{connected} \text{ EXCEPT } ![a] = \text{connected}[a] \setminus P] \\
&\wedge \text{UNCHANGED} \langle \text{msgs}, \text{maxBal}, \text{votesSent}, 2\text{avSent}, \text{received}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{FakeSend}(a) &\triangleq \\
&\wedge \exists m \in \{mm \in \text{Message} : \\
&\quad \wedge mm.\text{acc} = a \\
&\quad \wedge \vee mm.\text{type} = \text{"1b"} \\
&\quad \quad \vee mm.\text{type} = \text{"2av"} \\
&\quad \quad \vee mm.\text{type} = \text{"2b"}\} : \\
&\quad \text{Send}(m) \\
&\wedge \text{UNCHANGED} \langle \text{maxBal}, \text{votesSent}, 2\text{avSent}, \text{received}, \text{connected}, \text{receivedByLearner}, \text{decision} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{LearnerDecide}(l, b) &\triangleq \\
&\wedge \exists v \in \{vv \in \text{Value} : \text{ChosenIn}(l, b, vv)\} :
\end{aligned}$$

$$\begin{aligned}
& decision' = [decision \text{ EXCEPT } ![l, b] = decision[l, b] \cup \{v\}] \\
& \wedge \text{UNCHANGED } \langle msgs, maxBal, votesSent, 2avSent, received, connected, receivedByLearner \rangle \\
LearnerRecv(l) & \triangleq \\
& \exists m \in \{mm \in msgs : mm.type = \text{"2b"} \wedge mm.lr = l\} : \\
& \quad receivedByLearner' = \\
& \quad \quad [receivedByLearner \text{ EXCEPT } ![l] = receivedByLearner[l] \cup \{m\}] \\
& \wedge \text{UNCHANGED } \langle msgs, maxBal, votesSent, 2avSent, received, connected, decision \rangle \\
ProposerAction & \triangleq \\
& \exists lrn \in Learner : \exists proposer \in Ballot : \\
& \quad \vee Phase1a(lrn, proposer) \\
& \quad \vee \exists v \in Value : Phase1c(lrn, proposer, v) \\
AcceptorSendAction & \triangleq \\
& \exists lrn \in Learner : \exists bal \in Ballot : \exists acc \in SafeAcceptor : \exists val \in Value : \\
& \quad \vee Phase1b(lrn, bal, acc) \\
& \quad \vee Phase2av(lrn, bal, acc, val) \\
& \quad \vee Phase2b(lrn, bal, acc, val) \\
AcceptorReceiveAction & \triangleq \\
& \exists lrn \in Learner : \exists acc \in Acceptor : Recv(lrn, acc) \\
AcceptorDisconnectAction & \triangleq \\
& \exists acc \in SafeAcceptor : Disconnect(acc) \\
LearnerAction & \triangleq \\
& \exists lrn \in Learner : \\
& \quad \vee \exists bal \in Ballot : LearnerDecide(lrn, bal) \\
& \quad \vee LearnerRecv(lrn) \\
FakeAcceptorAction & \triangleq \exists a \in FakeAcceptor : FakeSend(a) \\
Next & \triangleq \\
& \vee ProposerAction \\
& \vee AcceptorSendAction \\
& \vee AcceptorReceiveAction \\
& \vee AcceptorDisconnectAction \\
& \vee LearnerAction \\
& \vee FakeAcceptorAction \\
Spec & \triangleq Init \wedge \Box[Next]_{vars}
\end{aligned}$$

$$\begin{aligned}
VotedFor(lr, acc, bal, val) & \triangleq \\
& \exists m \in msgs : \\
& \quad \wedge m.type = \text{"2b"}
\end{aligned}$$

$$\begin{aligned}
& \wedge m.lr = lr \\
& \wedge m.acc = acc \\
& \wedge m.bal = bal \\
& \wedge m.val = val
\end{aligned}$$

$$\begin{aligned}
Proposed(lr, acc, bal, val) & \triangleq \\
& \exists m \in msgs : \\
& \quad \wedge m.type = \text{"2av"} \\
& \quad \wedge m.lr = lr \\
& \quad \wedge m.acc = acc \\
& \quad \wedge m.bal = bal \\
& \quad \wedge m.val = val
\end{aligned}$$

$$\begin{aligned}
LeftBallot(lr, acc, bal) & \triangleq \\
& \exists m \in msgs : \\
& \quad \wedge m.type = \text{"1b"} \\
& \quad \wedge m.lr = lr \\
& \quad \wedge m.acc = acc \\
& \quad \wedge bal < m.bal
\end{aligned}$$

$$ReceivedSpec \triangleq \forall A \in SafeAcceptor : received[A] \subseteq msgs$$

$$\begin{aligned}
ReceivedByLearnerSpec & \triangleq \\
& \wedge receivedByLearner \in [Learner \rightarrow \text{SUBSET } \{mm \in msgs : mm.type = \text{"2b"}\}] \\
& \wedge \forall L \in Learner : \forall mm \in Message : \\
& \quad mm \in receivedByLearner[L] \Rightarrow mm.lr = L
\end{aligned}$$

$$\begin{aligned}
VotesSentSpec1 & \triangleq \\
& \forall A \in SafeAcceptor : \forall vote \in votesSent[A] : VotedFor(vote.lr, A, vote.bal, vote.val)
\end{aligned}$$

$$\begin{aligned}
VotesSentSpec2 & \triangleq \\
& \forall L \in Learner : \forall A \in SafeAcceptor : \forall B \in Ballot : \forall V \in Value : \\
& \quad VotedFor(L, A, B, V) \Rightarrow [lr \mapsto L, bal \mapsto B, val \mapsto V] \in votesSent[A]
\end{aligned}$$

$$\begin{aligned}
VotesSentSpec3 & \triangleq \\
& \forall A \in SafeAcceptor : \forall B \in Ballot : \forall vote \in votesSent[A] : \\
& \quad vote.bal < B \Rightarrow \\
& \quad \exists P \in votesSent[A] : \\
& \quad \quad MaxVote(A, B, P) \wedge P.lr = vote.lr \wedge vote.bal \leq P.bal
\end{aligned}$$

$$\begin{aligned}
VotesSentSpec4 & \triangleq \\
& \forall A \in SafeAcceptor : \forall vote1, vote2 \in votesSent[A] : \\
& \quad \langle vote1.lr, vote2.lr \rangle \in Ent \wedge \\
& \quad vote1.bal = vote2.bal \Rightarrow vote1.val = vote2.val
\end{aligned}$$

$$2avSentSpec1 \triangleq \forall A \in SafeAcceptor : \forall p \in 2avSent[A] : Proposed(p.lr, A, p.bal, p.val)$$

$$\begin{aligned}
2avSentSpec2 &\triangleq \\
&\forall L \in \text{Learner} : \forall A \in \text{SafeAcceptor} : \forall B \in \text{Ballot} : \forall V \in \text{Value} : \\
&\quad \text{Proposed}(L, A, B, V) \Rightarrow [lr \mapsto L, bal \mapsto B, val \mapsto V] \in 2avSent[A]
\end{aligned}$$

$$\begin{aligned}
2avSentSpec3 &\triangleq \\
&\forall L1, L2 \in \text{Learner} : \forall A \in \text{SafeAcceptor} : \forall B \in \text{Ballot} : \forall V1, V2 \in \text{Value} : \\
&\quad \langle L1, L2 \rangle \in \text{Ent} \wedge \\
&\quad [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in 2avSent[A] \wedge \\
&\quad [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent[A] \Rightarrow V1 = V2
\end{aligned}$$

$$\begin{aligned}
\text{ConnectedSpec} &\triangleq \\
&\forall A \in \text{SafeAcceptor} : \forall L1, L2 \in \text{Learner} : \\
&\quad \langle L1, L2 \rangle \in \text{Ent} \Rightarrow \langle L1, L2 \rangle \in \text{connected}[A]
\end{aligned}$$

$$\begin{aligned}
\text{DecisionSpec} &\triangleq \\
&\forall L \in \text{Learner} : \forall B \in \text{Ballot} : \forall V \in \text{Value} : \\
&\quad V \in \text{decision}[L, B] \Rightarrow \text{ChosenIn}(L, B, V)
\end{aligned}$$

$$\begin{aligned}
\text{MsgInv1b}(m) &\triangleq \\
&\wedge m.bal \leq \text{maxBal}[m.lr, m.acc] \\
&\wedge m.votes = \{p \in \text{votesSent}[m.acc] : \text{MaxVote}(m.acc, m.bal, p)\} \\
&\wedge m.proposals = \{p \in 2avSent[m.acc] : p.bal < m.bal \wedge p.lr = m.lr\}
\end{aligned}$$

$$\begin{aligned}
\text{MsgInv2av}(m) &\triangleq \\
&\wedge \text{InitializedBallot}(m.lr, m.bal) \\
&\wedge \text{AnnouncedValue}(m.lr, m.bal, m.val) \\
&\wedge \text{KnowsSafeAt}(m.lr, m.acc, m.bal, m.val) \\
&\wedge [lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in 2avSent[m.acc] \quad \text{TODO check if used} \\
&\wedge \exists Q \in \text{ByzQuorum} : \\
&\quad \wedge [lr \mapsto m.lr, q \mapsto Q] \in \text{TrustLive} \\
&\quad \wedge \forall ba \in Q : \\
&\quad \quad \exists m1b \in \text{received}[m.acc] : \\
&\quad \quad \quad \wedge m1b.type = \text{"1b"} \\
&\quad \quad \quad \wedge m1b.lr = m.lr \\
&\quad \quad \quad \wedge m1b.acc = ba \\
&\quad \quad \quad \wedge m1b.bal = m.bal
\end{aligned}$$

$$\begin{aligned}
\text{MsgInv2b}(m) &\triangleq \\
&\wedge [lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in \text{votesSent}[m.acc] \\
&\wedge \exists Q \in \text{ByzQuorum} : \\
&\quad \wedge [lr \mapsto m.lr, q \mapsto Q] \in \text{TrustLive} \\
&\quad \wedge \forall ba \in Q : \\
&\quad \quad \exists m2av \in \text{received}[m.acc] : \\
&\quad \quad \quad \wedge m2av.type = \text{"2av"} \\
&\quad \quad \quad \wedge m2av.lr = m.lr \\
&\quad \quad \quad \wedge m2av.acc = ba \\
&\quad \quad \quad \wedge m2av.bal = m.bal
\end{aligned}$$

$$\wedge m2av.val = m.val$$

$$\begin{aligned} MsgInv &\triangleq \forall m \in msgs : m.acc \in SafeAcceptor \Rightarrow \\ &\quad \wedge (m.type = \text{"1b"}) \Rightarrow MsgInv1b(m) \\ &\quad \wedge (m.type = \text{"2av"}) \Rightarrow MsgInv2av(m) \\ &\quad \wedge (m.type = \text{"2b"}) \Rightarrow MsgInv2b(m) \end{aligned}$$

LEMMA *MessageType* \triangleq

ASSUME NEW $m \in Message$

PROVE $\wedge m.lr \in Learner$

$\wedge m.bal \in Ballot$

$\wedge (m.type = \text{"1b"} \vee m.type = \text{"2av"} \vee m.type = \text{"2b"}) \Rightarrow m.acc \in Acceptor$

$\wedge (m.type = \text{"1c"} \vee m.type = \text{"2av"} \vee m.type = \text{"2b"}) \Rightarrow m.val \in Value$

$\wedge (m.type = \text{"1b"}) \Rightarrow$

$\wedge m.votes \in \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]$

$\wedge m.proposals \in \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]$

PROOF BY DEF *Message*

LEMMA *TypeOKInvariant* $\triangleq TypeOK \wedge Next \Rightarrow TypeOK'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME *TypeOK*, *Next* PROVE *TypeOK'* OBVIOUS

$\langle 1 \rangle$ USE DEF *Next*

$\langle 1 \rangle 1$. CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction*, *Phase1a*, *Phase1c*, *Send*, *TypeOK*, *Message*

$\langle 1 \rangle 2$. CASE *AcceptorSendAction*

$\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,

NEW $bal \in Ballot$,

NEW $acc \in Acceptor$,

NEW $val \in Value$,

$\vee Phase1b(lrn, bal, acc)$

$\vee Phase2av(lrn, bal, acc, val)$

$\vee Phase2b(lrn, bal, acc, val)$

PROVE *TypeOK'*

BY $\langle 1 \rangle 2$, *SafeAcceptorIsAcceptor* DEF *AcceptorSendAction*

$\langle 2 \rangle 1$. CASE *Phase1b*(lrn, bal, acc)

$\langle 3 \rangle 1$. ($votesSent \in [Acceptor \rightarrow \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]]$)'

BY $\langle 2 \rangle 1$ DEF *Phase1b*, *Phase2av*, *Phase2b*, *Send*, *TypeOK*, *Message*

$\langle 3 \rangle 2$. ($2avSent \in [Acceptor \rightarrow \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]]$)'

BY $\langle 2 \rangle 1$ DEF *Phase1b*, *Phase2av*, *Phase2b*, *Send*, *TypeOK*, *Message*

$\langle 3 \rangle 3$. $msgs' \in \text{SUBSET } Message$

$\langle 4 \rangle$ SUFFICES

$[type \mapsto \text{"1b"}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal,$

$votes \mapsto \{vote \in votesSent[acc] : MaxVote(acc, bal, vote)\},$

$proposals \mapsto \{p \in 2avSent[acc] : p.bal < bal \wedge p.lr = lrn\}] \in Message$

BY $\langle 2 \rangle 1$ DEF *Phase1b*, *Send*, *TypeOK*

$\langle 4 \rangle 1. \{ vote \in votesSent[acc] : MaxVote(acc, bal, vote) \}$
 $\in \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]$
 BY DEF *TypeOK*
 $\langle 4 \rangle 2. \{ p \in 2avSent[acc] : p.bal < bal \wedge p.lr = lrn \} \in \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]$
 BY DEF *TypeOK*
 $\langle 4 \rangle 3. \text{QED BY } \langle 4 \rangle 1, \langle 4 \rangle 2 \text{ DEF } Message, TypeOK$
 $\langle 3 \rangle 4. \text{QED BY } \langle 2 \rangle 1, \langle 3 \rangle 1, \langle 3 \rangle 2, \langle 3 \rangle 3 \text{ DEF } Phase1b, TypeOK, Send$
 $\langle 2 \rangle 2. \text{CASE } Phase2av(lrn, bal, acc, val)$
 $\langle 3 \rangle 2. msgs' \in \text{SUBSET } Message$
 $\langle 4 \rangle 0. [type \mapsto "2av", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val] \in Message$
 BY *SafeAcceptorIsAcceptor* DEF *Message*
 $\langle 4 \rangle 1. \text{QED BY } \langle 2 \rangle 2, \langle 4 \rangle 0, SafeAcceptorIsAcceptor \text{ DEF } Phase2av, Send, TypeOK, Message$
 $\langle 3 \rangle 4. (2avSent \in [Acceptor \rightarrow \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]])'$
 $\langle 4 \rangle 0. [lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \in [lr : Learner, bal : Ballot, val : Value]$
 BY DEF *TypeOK*
 $\langle 4 \rangle 1. \text{QED BY } \langle 2 \rangle 2, \langle 1 \rangle 2, \langle 4 \rangle 0, SafeAcceptorIsAcceptor \text{ DEF } Phase2av, Send, TypeOK, Message$
 $\langle 3 \rangle 5. \text{QED BY } \langle 2 \rangle 2, \langle 3 \rangle 2, \langle 3 \rangle 4 \text{ DEF } Phase2av, Send, TypeOK$
 $\langle 2 \rangle 3. \text{CASE } Phase2b(lrn, bal, acc, val)$
 $\langle 3 \rangle 1. val \in ValueOBVIOUS$
 $\langle 3 \rangle 2. msgs' \in \text{SUBSET } Message$
 $\langle 4 \rangle 0. [type \mapsto "2b", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val] \in Message$
 BY *SafeAcceptorIsAcceptor* DEF *Message*
 $\langle 4 \rangle 1. \text{QED BY } \langle 4 \rangle 0, \langle 2 \rangle 3 \text{ DEF } Phase2b, Message, Send, TypeOK$
 $\langle 3 \rangle 3. votesSent' \in [Acceptor \rightarrow \text{SUBSET } [lr : Learner, bal : Ballot, val : Value]]$
 $\langle 4 \rangle 0. [lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \in [lr : Learner, bal : Ballot, val : Value] \text{ BY } \langle 3 \rangle 1$
 $\langle 4 \rangle 1 \text{ QED BY } \langle 2 \rangle 3, \langle 1 \rangle 2, \langle 4 \rangle 0 \text{ DEF } Phase2b, TypeOK$
 $\langle 3 \rangle 5. \text{QED BY } \langle 2 \rangle 3, \langle 1 \rangle 2, \langle 3 \rangle 1, \langle 3 \rangle 2, \langle 3 \rangle 3 \text{ DEF } Phase2b, Send, TypeOK$
 $\langle 2 \rangle 4. \text{QED BY } \langle 1 \rangle 2, \langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3. \text{CASE } AcceptorReceiveAction$
 $\langle 2 \rangle \text{ SUFFICES ASSUME NEW } lrn \in Learner,$
 $\text{NEW } acc \in Acceptor,$
 $\text{NEW } m \in msgs,$
 $received' = [received \text{ EXCEPT } ![acc] = received[acc] \cup \{m\}],$
 $\text{UNCHANGED } \langle msgs, maxBal, 2avSent, votesSent, connected,$
 $\text{receivedByLearner, decision} \rangle$
 PROVE *TypeOK'*
 BY *SafeAcceptorIsAcceptor*, $\langle 1 \rangle 3 \text{ DEF } AcceptorReceiveAction, Recv$
 $\langle 2 \rangle 7. \text{QED BY } \langle 1 \rangle 3 \text{ DEF } AcceptorReceiveAction, Recv, TypeOK$
 $\langle 1 \rangle 4. \text{CASE } AcceptorDisconnectAction \text{ BY } \langle 1 \rangle 4 \text{ DEF } AcceptorDisconnectAction, Disconnect, TypeOK, Message$
 $\langle 1 \rangle 5. \text{CASE } LearnerAction$
 $\langle 2 \rangle 1. \text{ASSUME NEW } lrn \in Learner, \text{NEW } bal \in Ballot,$
 $LearnerDecide(lrn, bal)$
 PROVE *TypeOK'*
 BY $\langle 2 \rangle 1 \text{ DEF } LearnerDecide, TypeOK$
 $\langle 2 \rangle 2. \text{ASSUME NEW } lrn \in Learner, LearnerRecv(lrn)$

PROVE $TypeOK'$
 BY $\langle 2 \rangle 2$ DEF $LearnerRecv$, $TypeOK$
 $\langle 2 \rangle 3$. QED BY $\langle 1 \rangle 5$, $\langle 2 \rangle 1$, $\langle 2 \rangle 2$ DEF $LearnerAction$
 $\langle 1 \rangle 6$. CASE $FakeAcceptorAction$
 $\langle 2 \rangle 1$. SUFFICES ASSUME NEW $a \in Acceptor$, $FakeSend(a)$
 PROVE $TypeOK'$
 BY $\langle 1 \rangle 6$, $FakeAcceptorIsAcceptor$ DEF $FakeAcceptorAction$
 $\langle 2 \rangle 2$. QED BY $\langle 2 \rangle 1$ DEF $FakeSend$, $Send$, $TypeOK$
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$, $\langle 1 \rangle 4$, $\langle 1 \rangle 5$, $\langle 1 \rangle 6$ DEF $Next$

LEMMA $MsgsMonotone \triangleq Next \Rightarrow msgs \subseteq msgs'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME $Next$ PROVE $msgs \subseteq msgs'$ OBVIOUS
 $\langle 1 \rangle 1$. CASE $ProposerAction$ BY $\langle 1 \rangle 1$ DEF $ProposerAction$, $Phase1a$, $Phase1c$, $Send$
 $\langle 1 \rangle 2$. CASE $AcceptorSendAction$ BY $\langle 1 \rangle 2$ DEF $AcceptorSendAction$, $Phase1b$, $Phase2av$, $Phase2b$, $Send$
 $\langle 1 \rangle 3$. CASE $AcceptorReceiveAction$ BY $\langle 1 \rangle 3$ DEF $AcceptorReceiveAction$, $Recv$
 $\langle 1 \rangle 4$. CASE $AcceptorDisconnectAction$ BY $\langle 1 \rangle 4$ DEF $AcceptorDisconnectAction$, $Disconnect$
 $\langle 1 \rangle 5$. CASE $LearnerAction$ BY $\langle 1 \rangle 5$ DEF $LearnerAction$, $LearnerDecide$, $LearnerRecv$
 $\langle 1 \rangle 6$. CASE $FakeAcceptorAction$ BY $\langle 1 \rangle 6$ DEF $FakeAcceptorAction$, $FakeSend$, $Send$
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$, $\langle 1 \rangle 4$, $\langle 1 \rangle 5$, $\langle 1 \rangle 6$ DEF $Next$

LEMMA $ReceivedSpecInvariant \triangleq TypeOK \wedge ReceivedSpec \wedge Next \Rightarrow ReceivedSpec'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME $TypeOK$, $ReceivedSpec$, $Next$ PROVE $ReceivedSpec'$ OBVIOUS
 $\langle 1 \rangle 0$. $TypeOK'$ BY $TypeOKInvariant$
 $\langle 1 \rangle 1$. CASE $ProposerAction$
 BY $\langle 1 \rangle 1$, $SafeAcceptorIsAcceptor$ DEF $ProposerAction$, $Phase1a$, $Phase1c$, $ReceivedSpec$, $Send$, $Next$, $TypeOK$
 $\langle 1 \rangle 2$. CASE $AcceptorSendAction$
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,
 NEW $bal \in Ballot$,
 NEW $acc \in Acceptor$,
 NEW $val \in Value$,
 $\vee Phase1b(lrn, bal, acc)$
 $\vee Phase2av(lrn, bal, acc, val)$
 $\vee Phase2b(lrn, bal, acc, val)$
 PROVE $ReceivedSpec'$
 BY $\langle 1 \rangle 2$, $SafeAcceptorIsAcceptor$ DEF $AcceptorSendAction$
 $\langle 2 \rangle 1$. CASE $Phase1b(lrn, bal, acc)$ BY $\langle 2 \rangle 1$, $MsgsMonotone$ DEF $TypeOK$, $ReceivedSpec$, $Phase1b$
 $\langle 2 \rangle 2$. CASE $Phase2av(lrn, bal, acc, val)$ BY $\langle 2 \rangle 2$ DEF $TypeOK$, $ReceivedSpec$, $Phase2av$, $Send$
 $\langle 2 \rangle 3$. CASE $Phase2b(lrn, bal, acc, val)$ BY $\langle 2 \rangle 3$, $MsgsMonotone$ DEF $Phase2b$, $TypeOK$, $ReceivedSpec$, $Send$
 $\langle 2 \rangle 4$. QED BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$, $\langle 2 \rangle 3$
 $\langle 1 \rangle 3$. CASE $AcceptorReceiveAction$
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,
 NEW $acc \in Acceptor$,
 NEW $m \in msgs$,

$received' = [received \text{ EXCEPT } ![acc] = received[acc] \cup \{m\}],$
 UNCHANGED $\langle msgs, maxBal, 2avSent, votesSent, connected, receivedByLearner, decis \rangle$
 PROVE $ReceivedSpec'$
 BY $\langle 1 \rangle 3$, *SafeAcceptorIsAcceptor* DEF *AcceptorReceiveAction*, *Recv*
 $\langle 2 \rangle$ QED BY *MessageType*, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec*, *TypeOK*, *Next*
 $\langle 1 \rangle 4$.CASE *AcceptorDisconnectAction*
 BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction*, *Disconnect*, *ReceivedSpec*, *TypeOK*, *Next*
 $\langle 1 \rangle 5$.CASE *LearnerAction*
 BY $\langle 1 \rangle 5$ DEF *LearnerAction*, *LearnerRecv*, *LearnerDecide*, *ReceivedSpec*, *TypeOK*, *Next*
 $\langle 1 \rangle 6$.CASE *FakeAcceptorAction*
 $\langle 2 \rangle 1$. SUFFICES ASSUME NEW $a \in \text{Acceptor}$, *FakeSend*(a) PROVE $ReceivedSpec'$
 BY $\langle 1 \rangle 6$, *FakeAcceptorIsAcceptor* DEF *FakeAcceptorAction*
 $\langle 2 \rangle 2$. QED BY $\langle 2 \rangle 1$ DEF *FakeSend*, *Send*, *TypeOK*, *ReceivedSpec*
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$, $\langle 1 \rangle 4$, $\langle 1 \rangle 5$, $\langle 1 \rangle 6$ DEF *Next*

 LEMMA *ReceivedByLearnerSpecInvariant* \triangleq
 $TypeOK \wedge ReceivedByLearnerSpec \wedge Next \Rightarrow ReceivedByLearnerSpec'$
 PROOF
 $\langle 1 \rangle$ SUFFICES ASSUME *TypeOK*, *ReceivedByLearnerSpec*, *Next* PROVE $ReceivedByLearnerSpec'$ OBVIOUS
 $\langle 1 \rangle 1$.CASE *ProposerAction*
 BY $\langle 1 \rangle 1$ DEF *ProposerAction*, *Phase1a*, *Phase1c*, *ReceivedByLearnerSpec*, *Send*, *Next*, *TypeOK*
 $\langle 1 \rangle 2$.CASE *AcceptorSendAction*
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in \text{Learner}$,
 NEW $bal \in \text{Ballot}$,
 NEW $acc \in \text{Acceptor}$,
 NEW $val \in \text{Value}$,
 $\vee Phase1b(lrn, bal, acc)$
 $\vee Phase2av(lrn, bal, acc, val)$
 $\vee Phase2b(lrn, bal, acc, val)$
 PROVE $ReceivedByLearnerSpec'$
 BY $\langle 1 \rangle 2$, *SafeAcceptorIsAcceptor* DEF *AcceptorSendAction*
 $\langle 2 \rangle 1$.CASE *Phase1b*(lrn, bal, acc)
 BY $\langle 2 \rangle 1$ DEF *TypeOK*, *ReceivedByLearnerSpec*, *Phase1b*, *Send*
 $\langle 2 \rangle 2$.CASE *Phase2av*(lrn, bal, acc, val)
 BY $\langle 2 \rangle 2$ DEF *TypeOK*, *ReceivedByLearnerSpec*, *Phase2av*, *Send*
 $\langle 2 \rangle 3$.CASE *Phase2b*(lrn, bal, acc, val)
 $\langle 3 \rangle$ SUFFICES ASSUME $Send([type \mapsto "2b", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val])$
 PROVE $ReceivedByLearnerSpec'$
 BY $\langle 2 \rangle 3$ DEF *Phase2b*
 $\langle 3 \rangle 0$. *TypeOK'* BY *TypeOKInvariant*
 $\langle 3 \rangle 1$. UNCHANGED $\langle receivedByLearner \rangle$ BY $\langle 2 \rangle 3$ DEF *Phase2b*
 $\langle 3 \rangle 3$. $(\forall L \in \text{Learner} : \forall mm \in \text{Message} : mm \in receivedByLearner[L] \Rightarrow mm.lr = L)'$
 BY $\langle 3 \rangle 1$ DEF *ReceivedByLearnerSpec*, *TypeOK*
 $\langle 3 \rangle 4$. $(receivedByLearner \in [Learner \rightarrow \text{SUBSET } \{mm \in msgs : mm.type = "2b\}])'$
 BY $\langle 3 \rangle 0$, $\langle 3 \rangle 1$, *MessageType* DEF *ReceivedByLearnerSpec*, *Send*, *TypeOK*

$\langle 3 \rangle 5$. QED BY $\langle 3 \rangle 3, \langle 3 \rangle 4$ DEF *ReceivedByLearnerSpec*
 $\langle 2 \rangle 4$. QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3$. CASE *AcceptorReceiveAction*
BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction, Recv, ReceivedByLearnerSpec, TypeOK, Next*
 $\langle 1 \rangle 4$. CASE *AcceptorDisconnectAction*
BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction, Disconnect, ReceivedByLearnerSpec, TypeOK, Next*
 $\langle 1 \rangle 5$. CASE *LearnerAction*
 $\langle 2 \rangle 1$. ASSUME NEW $l_{rn} \in \text{Learner}$, NEW $bal \in \text{Ballot}$, *LearnerDecide*(l_{rn} , bal)
PROVE *ReceivedByLearnerSpec'*
BY $\langle 2 \rangle 1$ DEF *LearnerDecide, ReceivedByLearnerSpec, TypeOK, Next*
 $\langle 2 \rangle 2$. ASSUME NEW $l_{rn} \in \text{Learner}$, *LearnerRecv*(l_{rn})
PROVE *ReceivedByLearnerSpec'*
 $\langle 3 \rangle$ SUFFICES ASSUME NEW $m \in \{mm \in \text{msgs} : mm.type = \text{"2b"} \wedge mm.lr = l_{rn}\}$,
receivedByLearner' =
 $[receivedByLearner \text{ EXCEPT } ![l_{rn}] = receivedByLearner[l_{rn}] \cup \{m\}]$
PROVE *ReceivedByLearnerSpec'*
BY $\langle 2 \rangle 2$ DEF *LearnerRecv*
 $\langle 3 \rangle 1$. UNCHANGED $\langle \text{msgs} \rangle$ BY $\langle 2 \rangle 2$ DEF *LearnerAction, LearnerRecv*
 $\langle 3 \rangle 5$. QED BY $\langle 2 \rangle 2, \langle 3 \rangle 1$ DEF *ReceivedByLearnerSpec*
 $\langle 2 \rangle 3$. QED BY $\langle 1 \rangle 5, \langle 2 \rangle 1, \langle 2 \rangle 2$ DEF *LearnerAction*
 $\langle 1 \rangle 6$. CASE *FakeAcceptorAction*
 $\langle 2 \rangle 1$. SUFFICES ASSUME NEW $a \in \text{Acceptor}$, *FakeSend*(a) PROVE *ReceivedByLearnerSpec'*
BY $\langle 1 \rangle 6, \text{FakeAcceptorIsAcceptor}$ DEF *FakeAcceptorAction*
 $\langle 2 \rangle 2$. QED BY $\langle 2 \rangle 1$ DEF *FakeSend, Send, TypeOK, ReceivedByLearnerSpec*
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*
LEMMA *MaxBalMonotone* \triangleq
 $TypeOK \wedge Next \Rightarrow \forall l \in \text{Learner} : \forall a \in \text{SafeAcceptor} : maxBal[l, a] \leq maxBal'[l, a]$
PROOF
 $\langle 1 \rangle$ SUFFICES ASSUME *TypeOK, Next*, NEW CONSTANT $l \in \text{Learner}$, NEW CONSTANT $a \in \text{SafeAcceptor}$
PROVE $maxBal[l, a] \leq maxBal'[l, a]$
OBVIOUS
 $\langle 1 \rangle 1$. CASE *ProposerAction*
BY $\langle 1 \rangle 1, \text{SafeAcceptorIsAcceptor}$ DEF *ProposerAction, Phase1a, Phase1c, Send, TypeOK, Ballot*
 $\langle 1 \rangle 2$. CASE *AcceptorSendAction*
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $l_{rn} \in \text{Learner}$,
NEW $bal \in \text{Ballot}$,
NEW $acc \in \text{Acceptor}$,
NEW $val \in \text{Value}$,
 $\vee Phase1b(l_{rn}, bal, acc)$
 $\vee Phase2av(l_{rn}, bal, acc, val)$
 $\vee Phase2b(l_{rn}, bal, acc, val)$
PROVE $maxBal[l, a] \leq (maxBal')[l, a]$
BY $\langle 1 \rangle 2, \text{SafeAcceptorIsAcceptor}$ DEF *AcceptorSendAction*
 $\langle 2 \rangle 1$. CASE *Phase1b*(l_{rn}, bal, acc)

$\langle 3 \rangle 1.$ CASE $\langle l, a \rangle = \langle l_{rn}, acc \rangle$ BY $\langle 2 \rangle 1, \langle 3 \rangle 1$ DEF *Phase1b, TypeOK, Ballot*
 $\langle 3 \rangle 2.$ CASE $\langle l, a \rangle \neq \langle l_{rn}, acc \rangle$ BY $\langle 2 \rangle 1, \langle 3 \rangle 2, SafeAcceptorIsAcceptor$ DEF *Phase1b, TypeOK, Ballot*
 $\langle 3 \rangle 3.$ QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2$
 $\langle 2 \rangle 2.$ CASE *Phase2av*(*l_{rn}, bal, acc, val*)
 $\langle 3 \rangle 1.$ UNCHANGED *maxBal* BY $\langle 2 \rangle 2$ DEF *Phase2av*
 $\langle 3 \rangle 2.$ QED BY $\langle 3 \rangle 1, SafeAcceptorIsAcceptor$ DEF *TypeOK, Ballot*
 $\langle 2 \rangle 3.$ CASE *Phase2b*(*l_{rn}, bal, acc, val*)
 $\langle 3 \rangle 1.$ UNCHANGED *maxBal* BY $\langle 2 \rangle 3$ DEF *Phase2b*
 $\langle 3 \rangle 2.$ QED BY $\langle 3 \rangle 1, SafeAcceptorIsAcceptor$ DEF *TypeOK, Ballot*
 $\langle 2 \rangle 4.$ QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3.$ CASE *AcceptorReceiveAction*
 $\langle 2 \rangle 1.$ UNCHANGED *maxBal* BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction, Recv*
 $\langle 2 \rangle 2.$ QED BY $\langle 2 \rangle 1, SafeAcceptorIsAcceptor$ DEF *TypeOK, Ballot*
 $\langle 1 \rangle 4.$ CASE *AcceptorDisconnectAction*
 $\langle 2 \rangle 1.$ UNCHANGED *maxBal* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction, Disconnect*
 $\langle 2 \rangle 2.$ QED BY $\langle 2 \rangle 1, SafeAcceptorIsAcceptor$ DEF *TypeOK, Ballot*
 $\langle 1 \rangle 5.$ CASE *LearnerAction*
 $\langle 2 \rangle 1.$ UNCHANGED *maxBal* BY $\langle 1 \rangle 5$ DEF *LearnerAction, LearnerDecide, LearnerRecv*
 $\langle 2 \rangle 2.$ QED BY $\langle 2 \rangle 1, SafeAcceptorIsAcceptor$ DEF *TypeOK, Ballot*
 $\langle 1 \rangle 6.$ CASE *FakeAcceptorAction*
 $\langle 2 \rangle 1.$ UNCHANGED *maxBal* BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction, FakeSend*
 $\langle 2 \rangle 2.$ QED BY $\langle 2 \rangle 1, SafeAcceptorIsAcceptor$ DEF *TypeOK, Ballot*
 $\langle 1 \rangle 7.$ QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

LEMMA *2avSentMonotone* $\triangleq TypeOK \wedge Next \Rightarrow \forall A \in SafeAcceptor : 2avSent[A] \subseteq 2avSent'[A]$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME *TypeOK, Next*, NEW $A \in SafeAcceptor$ PROVE $2avSent[A] \subseteq 2avSent[A]'$ OBVIOUS

$\langle 1 \rangle 0a.$ *TypeOK* OBVIOUS

$\langle 1 \rangle 0b.$ *TypeOK'* BY *TypeOKInvariant*

$\langle 1 \rangle 1.$ CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction, Phase1a, Phase1c, Send*

$\langle 1 \rangle 2.$ CASE *AcceptorSendAction*

$\langle 2 \rangle$ SUFFICES ASSUME NEW *l_{rn} ∈ Learner*,
NEW *bal ∈ Ballot*,
NEW *acc ∈ Acceptor*,
NEW *val ∈ Value*,
 $\vee Phase1b(l_{rn}, bal, acc)$
 $\vee Phase2av(l_{rn}, bal, acc, val)$
 $\vee Phase2b(l_{rn}, bal, acc, val)$

PROVE $2avSent[A] \subseteq 2avSent[A]'$

BY $\langle 1 \rangle 2, SafeAcceptorIsAcceptor$ DEF *AcceptorSendAction*

$\langle 2 \rangle 1.$ QED BY $\langle 1 \rangle 0b, SafeAcceptorIsAcceptor$ DEF *AcceptorSendAction, Phase1b, Phase2av, Phase2b, Send*,

$\langle 1 \rangle 3.$ CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction, Recv*

$\langle 1 \rangle 4.$ CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction, Disconnect*

$\langle 1 \rangle 5.$ CASE *LearnerAction* BY $\langle 1 \rangle 5$ DEF *LearnerAction, LearnerDecide, LearnerRecv*

$\langle 1 \rangle 6.$ CASE *FakeAcceptorAction* BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction, FakeSend, Send*

$\langle 1 \rangle 7.$ QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

LEMMA *ReceivedMonotone* \triangleq

$TypeOK \wedge Next \Rightarrow \forall A \in SafeAcceptor : received[A] \subseteq received'[A]$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME *TypeOK*, *Next*, NEW $A \in SafeAcceptor$

PROVE $received[A] \subseteq received'[A]$ OBVIOUS

$\langle 1 \rangle 0a.$ *TypeOK* OBVIOUS

$\langle 1 \rangle 0b.$ *TypeOK'* BY *TypeOKInvariant*

$\langle 1 \rangle 1.$ CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction*, *Phase1a*, *Phase1c*, *Send*

$\langle 1 \rangle 2.$ CASE *AcceptorSendAction* BY $\langle 1 \rangle 2$ DEF *AcceptorSendAction*, *Send*, *Phase1b*, *Phase2av*, *Phase2b*

$\langle 1 \rangle 3.$ CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 3, \langle 1 \rangle 0a, \langle 1 \rangle 0b$, *SafeAcceptorIsAcceptor* DEF *AcceptorReceiveAction*, *Recv*, *TypeOK*

$\langle 1 \rangle 4.$ CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction*, *Disconnect*

$\langle 1 \rangle 5.$ CASE *LearnerAction* BY $\langle 1 \rangle 5$ DEF *LearnerAction*, *LearnerDecide*, *LearnerRecv*

$\langle 1 \rangle 6.$ CASE *FakeAcceptorAction* BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction*, *FakeSend*, *Send*

$\langle 1 \rangle 7.$ QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

LEMMA *VotesSentMonotone* \triangleq

$TypeOK \wedge Next \Rightarrow \forall A \in Acceptor : votesSent[A] \subseteq votesSent'[A]$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME *TypeOK*, *Next*, NEW $A \in Acceptor$ PROVE $votesSent[A] \subseteq votesSent'[A]$ OBVIOUS

$\langle 1 \rangle 0a.$ *TypeOK* OBVIOUS

$\langle 1 \rangle 0b.$ *TypeOK'* BY *TypeOKInvariant*

$\langle 1 \rangle 1.$ CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction*, *Phase1a*, *Phase1c*, *Send*

$\langle 1 \rangle 2.$ CASE *AcceptorSendAction* BY $\langle 1 \rangle 2, \langle 1 \rangle 0a, \langle 1 \rangle 0b$ DEF *AcceptorSendAction*, *Send*, *Phase1b*, *Phase2av*, *Phase2b*

$\langle 1 \rangle 3.$ CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 3, \langle 1 \rangle 0a, \langle 1 \rangle 0b$ DEF *AcceptorReceiveAction*, *Recv*, *TypeOK*

$\langle 1 \rangle 4.$ CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction*, *Disconnect*

$\langle 1 \rangle 5.$ CASE *LearnerAction* BY $\langle 1 \rangle 5$ DEF *LearnerAction*, *LearnerDecide*, *LearnerRecv*

$\langle 1 \rangle 6.$ CASE *FakeAcceptorAction* BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction*, *FakeSend*, *Send*

$\langle 1 \rangle 7.$ QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

LEMMA *VotesSentSpec1Invariant* $\triangleq Next \wedge VotesSentSpec1 \Rightarrow VotesSentSpec1'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME

Next, *VotesSentSpec1*, NEW $A \in SafeAcceptor$, NEW $vote \in votesSent'[A]$

PROVE *VotedFor*(*vote.lr*, *A*, *vote.bal*, *vote.val*)'

BY DEF *VotesSentSpec1*

$\langle 1 \rangle$ USE DEF *VotesSentSpec1*

$\langle 1 \rangle 1.$ CASE *ProposerAction* BY $\langle 1 \rangle 1$, *SafeAcceptorIsAcceptor* DEF *ProposerAction*, *Phase1a*, *Phase1c*, *Next*, *Send*

$\langle 1 \rangle 2.$ CASE *AcceptorSendAction*

$\langle 2 \rangle.$ SUFFICES ASSUME NEW *lrn* \in *Learner*,

NEW *bal* \in *Ballot*,

NEW *acc* \in *SafeAcceptor*,

NEW *val* \in *Value*,

$\vee Phase1b(lrn, bal, acc)$

$\vee Phase2av(lrn, bal, acc, val)$

$\vee \text{Phase2b}(lrn, bal, acc, val)$
 PROVE $\text{VotedFor}(\text{vote}.lr, A, \text{vote}.bal, \text{vote}.val)'$
 BY $\langle 1 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 2 \rangle 1$. CASE *Phase1b*(lrn, bal, acc) BY $\langle 2 \rangle 1$ DEF *Phase1b*
 $\langle 2 \rangle 2$. CASE *Phase2av*(lrn, bal, acc, val) BY $\langle 2 \rangle 2$ DEF *Phase2av*
 $\langle 2 \rangle 3$. CASE *Phase2b*(lrn, bal, acc, val)
 $\langle 3 \rangle$ SUFFICES ASSUME $\text{Send}([type \mapsto "2b", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]),$
 $\text{votesSent}' = [\text{votesSent} \text{ EXCEPT } ![acc] =$
 $\text{votesSent}[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\})$
 PROVE $\text{VotedFor}(\text{vote}.lr, A, \text{vote}.bal, \text{vote}.val)'$
 BY $\langle 2 \rangle 3$ DEF *Phase2b*
 $\langle 3 \rangle 2$. CASE $acc = A$
 $\langle 4 \rangle 1$. USE DEF *VotedFor*
 $\langle 4 \rangle 2$. CASE $\text{vote} \in \text{votesSent}[acc]$ BY $\langle 3 \rangle 2, \langle 4 \rangle 2, \text{MsgsMonotone}$
 $\langle 4 \rangle 3$. CASE $\text{vote} \notin \text{votesSent}[acc]$
 $\langle 5 \rangle 1$. DEFINE $m0 \triangleq [type \mapsto "2b", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]$
 $\langle 5 \rangle 2$. $m0 \in \text{msgs}'$ BY DEF *Phase2b, Send*
 $\langle 5 \rangle 3$. WITNESS $\langle 5 \rangle 2$
 $\langle 5 \rangle 10$ QED BY $\langle 3 \rangle 2, \langle 4 \rangle 3$
 $\langle 4 \rangle 4$. QED BY $\langle 4 \rangle 2, \langle 4 \rangle 3$
 $\langle 3 \rangle 3$. CASE $acc \neq A$ BY $\langle 3 \rangle 3$
 $\langle 3 \rangle 4$ QED BY $\langle 3 \rangle 2, \langle 3 \rangle 3$
 $\langle 2 \rangle 5$. QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3$. CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction, Recv, Next*
 $\langle 1 \rangle 4$. CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction, Disconnect, Next*
 $\langle 1 \rangle 5$. CASE *LearnerAction* BY $\langle 1 \rangle 5$ DEF *LearnerAction, LearnerRecv, LearnerDecide, Next*
 $\langle 1 \rangle 6$. CASE *FakeAcceptorAction* BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction, FakeSend, Send*
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*
 LEMMA $\text{VotesSentSpec2Invariant} \triangleq \text{TypeOK} \wedge \text{Next} \wedge \text{VotesSentSpec2} \Rightarrow \text{VotesSentSpec2}'$
 PROOF
 $\langle 1 \rangle$ SUFFICES ASSUME $\text{TypeOK}, \text{Next}, \text{VotesSentSpec2},$
 NEW $L \in \text{Learner}, \text{NEW } A \in \text{SafeAcceptor}, \text{NEW } B \in \text{Ballot}, \text{NEW } V \in \text{Value}$
 PROVE $(\text{VotedFor}(L, A, B, V) \Rightarrow [lr \mapsto L, bal \mapsto B, val \mapsto V] \in \text{votesSent}[A])'$
 BY DEF *VotesSentSpec2*
 $\langle 1 \rangle$ USE DEF *VotesSentSpec2*
 $\langle 1 \rangle 0a$. *TypeOK* OBVIOUS
 $\langle 1 \rangle 0b$. *TypeOK'* BY *TypeOKInvariant*
 $\langle 1 \rangle 1$. CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction, Phase1a, Phase1c, Send*
 $\langle 1 \rangle 2$. CASE *AcceptorSendAction*
 $\langle 2 \rangle$. SUFFICES ASSUME NEW $lrn \in \text{Learner},$
 NEW $bal \in \text{Ballot},$
 NEW $acc \in \text{SafeAcceptor},$
 NEW $val \in \text{Value},$
 $\vee \text{Phase1b}(lrn, bal, acc)$

$\vee \text{Phase2av}(\text{lrn}, \text{bal}, \text{acc}, \text{val})$
 $\vee \text{Phase2b}(\text{lrn}, \text{bal}, \text{acc}, \text{val})$
 PROVE $(\text{VotedFor}(L, A, B, V) \Rightarrow [\text{lr} \mapsto L, \text{bal} \mapsto B, \text{val} \mapsto V] \in \text{votesSent}[A])'$
 BY $\langle 1 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 2 \rangle 1$.CASE *Phase1b*(*lrn*, *bal*, *acc*)BY $\langle 2 \rangle 1$ DEF *Phase1b*
 $\langle 2 \rangle 2$.CASE *Phase2av*(*lrn*, *bal*, *acc*, *val*)BY $\langle 2 \rangle 2$ DEF *Phase2av*
 $\langle 2 \rangle 3$.CASE *Phase2b*(*lrn*, *bal*, *acc*, *val*)
 $\langle 3 \rangle$ SUFFICES ASSUME *Send*($[\text{type} \mapsto \text{"2b"}, \text{lr} \mapsto \text{lrn}, \text{acc} \mapsto \text{acc}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]$),
 $\text{votesSent}' = [\text{votesSent} \text{ EXCEPT } ![\text{acc}] =$
 $\text{votesSent}[\text{acc}] \cup \{[\text{lr} \mapsto \text{lrn}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]\}]$
 PROVE $(\text{VotedFor}(L, A, B, V) \Rightarrow [\text{lr} \mapsto L, \text{bal} \mapsto B, \text{val} \mapsto V] \in \text{votesSent}[A])'$
 BY $\langle 2 \rangle 3$ DEF *Phase2b*
 $\langle 3 \rangle 1$. QED BY $\langle 1 \rangle 0b$ DEF *Send*, *VotedFor*, *TypeOK*
 $\langle 2 \rangle 5$. QED BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$, $\langle 2 \rangle 3$
 $\langle 1 \rangle 3$.CASE *AcceptorReceiveAction*BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction*, *Recv*, *Next*
 $\langle 1 \rangle 4$.CASE *AcceptorDisconnectAction*BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction*, *Disconnect*, *Next*
 $\langle 1 \rangle 5$.CASE *LearnerAction*BY $\langle 1 \rangle 5$ DEF *LearnerAction*, *LearnerRecv*, *LearnerDecide*, *Next*
 $\langle 1 \rangle 6$.CASE *FakeAcceptorAction*BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction*, *FakeSend*, *Send*
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$, $\langle 1 \rangle 4$, $\langle 1 \rangle 5$, $\langle 1 \rangle 6$ DEF *Next*

 LEMMA *VotesSentSpec3Invariant* $\triangleq \text{TypeOK} \wedge \text{Next} \wedge \text{VotesSentSpec3} \Rightarrow \text{VotesSentSpec3}'$
 PROOF
 $\langle 1 \rangle$ SUFFICES ASSUME *TypeOK*, *Next*, *VotesSentSpec3*,
 NEW *A* $\in \text{SafeAcceptor}$, NEW *B* $\in \text{Ballot}$,
 NEW *V* $\in \text{votesSent}'[A]$,
 $V.\text{bal} < B$
 PROVE $(\exists P \in \text{votesSent}[A] : \text{MaxVote}(A, B, P) \wedge P.\text{lr} = V.\text{lr} \wedge V.\text{bal} \leq P.\text{bal})'$
 BY DEF *VotesSentSpec3*
 $\langle 1 \rangle$ USE DEF *VotesSentSpec3*
 $\langle 1 \rangle 0a$. *TypeOK* OBVIOUS
 $\langle 1 \rangle 0b$. *TypeOK'* BY *TypeOKInvariant*
 $\langle 1 \rangle 1$.CASE *ProposerAction*BY $\langle 1 \rangle 1$ DEF *ProposerAction*, *Phase1a*, *Phase1c*, *Send*
 $\langle 1 \rangle 2$.CASE *AcceptorSendAction*
 $\langle 2 \rangle$.SUFFICES ASSUME NEW *lrn* $\in \text{Learner}$,
 NEW *bal* $\in \text{Ballot}$,
 NEW *acc* $\in \text{SafeAcceptor}$,
 NEW *val* $\in \text{Value}$,
 $\vee \text{Phase1b}(\text{lrn}, \text{bal}, \text{acc})$
 $\vee \text{Phase2av}(\text{lrn}, \text{bal}, \text{acc}, \text{val})$
 $\vee \text{Phase2b}(\text{lrn}, \text{bal}, \text{acc}, \text{val})$
 PROVE $(\exists P \in \text{votesSent}[A] : \text{MaxVote}(A, B, P) \wedge P.\text{lr} = V.\text{lr} \wedge V.\text{bal} \leq P.\text{bal})'$
 BY $\langle 1 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 2 \rangle 1$.CASE *Phase1b*(*lrn*, *bal*, *acc*)BY $\langle 2 \rangle 1$ DEF *Phase1b*
 $\langle 2 \rangle 2$.CASE *Phase2av*(*lrn*, *bal*, *acc*, *val*)BY $\langle 2 \rangle 2$ DEF *Phase2av*
 $\langle 2 \rangle 3$.CASE *Phase2b*(*lrn*, *bal*, *acc*, *val*)

$\langle 3 \rangle$ SUFFICES ASSUME $votesSent' = [votesSent \text{ EXCEPT } ![acc] =$
 $votesSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}$
PROVE $(\exists P \in votesSent[A] : MaxVote(A, B, P) \wedge P.lr = V.lr \wedge V.bal \leq P.bal)'$
BY $\langle 2 \rangle 3$ DEF *Phase2b*
 $\langle 3 \rangle 1$. CASE $A = acc$
 $\langle 4 \rangle 0$. DEFINE $v0 \triangleq [lr \mapsto lrn, bal \mapsto bal, val \mapsto val]$
 $\langle 4 \rangle 1$. $v0 \in votesSent[A]'$ BY $\langle 3 \rangle 1, \langle 1 \rangle 0b$ DEF *TypeOK*
 $\langle 4 \rangle 2$. CASE $V \in votesSent[A]$ BY $\langle 4 \rangle 2$
 $\langle 4 \rangle 3$. CASE $V \notin votesSent[A]$
 $\langle 5 \rangle 0$. $V = v0$ BY $\langle 4 \rangle 3, \langle 3 \rangle 1$
 $\langle 5 \rangle 1$. CASE $\forall P \in votesSent[A] : P.lr = lrn \Rightarrow P.bal \geq B$
 $\langle 6 \rangle 1$. WITNESS $v0 \in votesSent[A]'$
 $\langle 6 \rangle 2$. QED BY $\langle 3 \rangle 1, \langle 5 \rangle 1, \langle 1 \rangle 0b, \langle 5 \rangle 0$ DEF *Ballot, TypeOK, MaxVote*
 $\langle 5 \rangle 2$. CASE $\exists P \in votesSent[A] : P.lr = lrn \wedge P.bal < B$
 $\langle 6 \rangle 1$. PICK $P \in votesSent[A] : P.lr = lrn \wedge P.bal < B$ BY $\langle 5 \rangle 2$
 $\langle 6 \rangle 2$. PICK $Pmax \in votesSent[A] : MaxVote(A, B, Pmax) \wedge Pmax.lr = lrn \wedge P.bal \leq Pmax.bal$ BY $\langle 6 \rangle 1$
 $\langle 6 \rangle 3$. $Pmax \in votesSent[A]'$ BY $\langle 3 \rangle 1, \langle 6 \rangle 2$
 $\langle 6 \rangle 4$. CASE $Pmax.bal < bal$
 $\langle 7 \rangle 1$. WITNESS $v0 \in votesSent[A]'$
 $\langle 7 \rangle 2$. SUFFICES $MaxVote(A, B, v0)'$ BY $\langle 5 \rangle 0$ DEF *Ballot*
 $\langle 7 \rangle 3$. QED BY $\langle 5 \rangle 0, \langle 6 \rangle 4, \langle 6 \rangle 2, \langle 1 \rangle 0b, \langle 3 \rangle 1$ DEF *Ballot, TypeOK, MaxVote*
 $\langle 6 \rangle 5$. CASE $bal \leq Pmax.bal$
 $\langle 7 \rangle 1$. WITNESS $Pmax \in votesSent[A]'$
 $\langle 7 \rangle 20$. QED BY $\langle 5 \rangle 0, \langle 6 \rangle 5, \langle 6 \rangle 2, \langle 1 \rangle 0b, \langle 3 \rangle 1$ DEF *Ballot, TypeOK*
 $\langle 6 \rangle 20$. QED BY $\langle 6 \rangle 4, \langle 6 \rangle 5$ DEF *Ballot, TypeOK*
 $\langle 5 \rangle 3$. QED BY $\langle 5 \rangle 1, \langle 5 \rangle 2$ DEF *Ballot, TypeOK*
 $\langle 4 \rangle 4$. QED BY $\langle 4 \rangle 2, \langle 4 \rangle 3$
 $\langle 3 \rangle 2$. CASE $A \neq acc$ BY $\langle 3 \rangle 2$
 $\langle 3 \rangle 3$. QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2$
 $\langle 2 \rangle 5$. QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3$. CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction, Recv, Next*
 $\langle 1 \rangle 4$. CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction, Disconnect, Next*
 $\langle 1 \rangle 5$. CASE *LearnerAction* BY $\langle 1 \rangle 5$ DEF *LearnerAction, LearnerRecv, LearnerDecide, Next*
 $\langle 1 \rangle 6$. CASE *FakeAcceptorAction* BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction, FakeSend, Send*
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*
LEMMA *VotesSentSpec4Invariant* \triangleq
TypeOK \wedge *Next* \wedge *MsgInv* \wedge *ReceivedSpec* \wedge
VotesSentSpec1 \wedge *2avSentSpec2* \wedge *2avSentSpec3* \wedge *VotesSentSpec4* \Rightarrow
VotesSentSpec4'
PROOF
 $\langle 1 \rangle$ SUFFICES ASSUME *TypeOK, Next, MsgInv, ReceivedSpec, VotesSentSpec1,*
2avSentSpec2, 2avSentSpec3, VotesSentSpec4,
NEW $A \in SafeAcceptor,$
NEW $vote1 \in votesSent'[A],$ NEW $vote2 \in votesSent'[A],$

$$\begin{array}{l}
\langle \text{vote1.lr}, \text{vote2.lr} \rangle \in \text{Ent}, \\
\text{vote1.bal} = \text{vote2.bal} \\
\text{PROVE } \text{vote1.val} = \text{vote2.val} \\
\text{BY DEF VotesSentSpec4} \\
\langle 1 \rangle \text{ USE DEF MsgInv} \\
\langle 1 \rangle 0a. \text{TypeOK OBVIOUS} \\
\langle 1 \rangle 0b. \text{TypeOK' BY TypeOKInvariant} \\
\langle 1 \rangle 1. \text{CASE ProposerAction BY } \langle 1 \rangle 1 \text{ DEF ProposerAction, Phase1a, Phase1c, Send, VotesSentSpec4} \\
\langle 1 \rangle 2. \text{CASE AcceptorSendAction} \\
\langle 2 \rangle. \text{SUFFICES ASSUME NEW } \text{lrn} \in \text{Learner}, \\
\quad \text{NEW } \text{bal} \in \text{Ballot}, \\
\quad \text{NEW } \text{acc} \in \text{SafeAcceptor}, \\
\quad \text{NEW } \text{val} \in \text{Value}, \\
\quad \vee \text{Phase1b}(\text{lrn}, \text{bal}, \text{acc}) \\
\quad \vee \text{Phase2av}(\text{lrn}, \text{bal}, \text{acc}, \text{val}) \\
\quad \vee \text{Phase2b}(\text{lrn}, \text{bal}, \text{acc}, \text{val}) \\
\text{PROVE } \text{vote1.val} = \text{vote2.val} \\
\text{BY } \langle 1 \rangle 2 \text{ DEF AcceptorSendAction} \\
\langle 2 \rangle 1. \text{CASE Phase1b}(\text{lrn}, \text{bal}, \text{acc}) \text{ BY } \langle 2 \rangle 1 \text{ DEF Phase1b, VotesSentSpec4} \\
\langle 2 \rangle 2. \text{CASE Phase2av}(\text{lrn}, \text{bal}, \text{acc}, \text{val}) \text{ BY } \langle 2 \rangle 2 \text{ DEF Phase2av, VotesSentSpec4} \\
\langle 2 \rangle 3. \text{CASE Phase2b}(\text{lrn}, \text{bal}, \text{acc}, \text{val}) \\
\langle 3 \rangle \text{ SUFFICES ASSUME } \text{votesSent}' = [\text{votesSent} \text{ EXCEPT } ![\text{acc}] = \\
\quad \text{votesSent}[\text{acc}] \cup \{[\text{lr} \mapsto \text{lrn}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]\}] \\
\text{PROVE } \text{vote1.val} = \text{vote2.val} \\
\text{BY } \langle 2 \rangle 3 \text{ DEF Phase2b} \\
\langle 3 \rangle 1. \text{CASE } A = \text{acc} \\
\langle 4 \rangle 1. \text{CASE } \text{vote1} \in \text{votesSent}[A] \wedge \text{vote2} \in \text{votesSent}[A] \text{ BY } \langle 4 \rangle 1 \text{ DEF VotesSentSpec4} \\
\langle 4 \rangle 2. \text{CASE } \text{vote1} \in \text{votesSent}[A] \wedge \text{vote2} \notin \text{votesSent}[A] \\
\langle 5 \rangle 0. \text{vote1.lr} \in \text{Learner} \wedge \text{vote1.val} \in \text{Value} \text{ BY } \langle 4 \rangle 2 \text{ DEF TypeOK} \\
\langle 5 \rangle 1. \text{vote2} = [\text{lr} \mapsto \text{lrn}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}] \text{ BY } \langle 4 \rangle 2 \\
\langle 5 \rangle 2. \text{PICK } Q2 \in \text{ByzQuorum} : \\
\quad \wedge [\text{lr} \mapsto \text{lrn}, q \mapsto Q2] \in \text{TrustLive} \\
\quad \wedge \forall aa \in Q2 : \\
\quad \quad \exists m \in \{mm \in \text{received}[\text{acc}] : \\
\quad \quad \quad \wedge mm.\text{type} = \text{"2av"} \\
\quad \quad \quad \wedge mm.\text{lr} = \text{lrn} \\
\quad \quad \quad \wedge mm.\text{bal} = \text{bal}\} : \\
\quad \quad \wedge m.\text{val} = \text{val} \\
\quad \quad \wedge m.\text{acc} = aa \\
\text{BY } \langle 5 \rangle 1, \langle 2 \rangle 3 \text{ DEF Phase2b} \\
\langle 5 \rangle 3. \langle \text{vote1.lr}, \text{lrn} \rangle \in \text{Ent} \wedge \text{vote1.bal} = \text{bal} \text{ BY } \langle 5 \rangle 1 \\
\langle 5 \rangle 4. \text{PICK } m1 \in \text{msgs} : \\
\quad \wedge m1.\text{type} = \text{"2b"} \\
\quad \wedge m1.\text{lr} = \text{vote1.lr} \\
\quad \wedge m1.\text{acc} = A
\end{array}$$

$\wedge m1.bal = bal$
 $\wedge m1.val = vote1.val$
 BY $\langle 4 \rangle 2, \langle 5 \rangle 3$ DEF *VotesSentSpec1, VotedFor*
 $\langle 5 \rangle 5$. PICK $Q1 \in ByzQuorum$:
 $\wedge [lr \mapsto vote1.lr, q \mapsto Q1] \in TrustLive$
 $\wedge \forall ba \in Q1$:
 $\exists m2av \in received[acc]$:
 $\wedge m2av.type = "2av"$
 $\wedge m2av.lr = vote1.lr$
 $\wedge m2av.acc = ba$
 $\wedge m2av.bal = bal$
 $\wedge m2av.val = vote1.val$
 $\langle 6 \rangle 1$. $\exists Q1 \in ByzQuorum$:
 $\wedge [lr \mapsto m1.lr, q \mapsto Q1] \in TrustLive$
 $\wedge \forall ba \in Q1$:
 $\exists m2av \in received[m1.acc]$:
 $\wedge m2av.type = "2av"$
 $\wedge m2av.lr = m1.lr$
 $\wedge m2av.acc = ba$
 $\wedge m2av.bal = m1.bal$
 $\wedge m2av.val = m1.val$
 BY $\langle 5 \rangle 4, \langle 3 \rangle 1$ DEF *MsgInv2b, TypeOK*
 $\langle 6 \rangle 2$. QED BY $\langle 5 \rangle 4, \langle 6 \rangle 1, \langle 3 \rangle 1$
 $\langle 5 \rangle 6$. $\langle vote1.lr, lrn \rangle \in EntBY \langle 5 \rangle 3$
 $\langle 5 \rangle 7$. PICK $S \in SafeAcceptor$: $S \in Q1 \wedge S \in Q2$ BY $\langle 5 \rangle 2, \langle 5 \rangle 5, \langle 5 \rangle 6, \langle 5 \rangle 0$, *EntanglementTrustLive*
 $\langle 5 \rangle 8$. PICK $m2av1 \in received[acc]$:
 $\wedge m2av1.type = "2av"$
 $\wedge m2av1.lr = vote1.lr$
 $\wedge m2av1.acc = S$
 $\wedge m2av1.bal = bal$
 $\wedge m2av1.val = vote1.val$
 BY $\langle 5 \rangle 7, \langle 5 \rangle 5$
 $\langle 5 \rangle 9$. $\wedge m2av1 \in msgs$
 $\wedge m2av1.type = "2av"$
 $\wedge m2av1.lr = vote1.lr$
 $\wedge m2av1.acc = S$
 $\wedge m2av1.bal = bal$
 $\wedge m2av1.val = vote1.val$
 BY $\langle 5 \rangle 8, \langle 5 \rangle 0$, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec, TypeOK*
 $\langle 5 \rangle 10$. $[lr \mapsto vote1.lr, bal \mapsto bal, val \mapsto vote1.val] \in 2avSent[S]$
 BY $\langle 5 \rangle 9, \langle 5 \rangle 0$ DEF *2avSentSpec2, Proposed*
 $\langle 5 \rangle 11$. PICK $m2av2 \in received[acc]$:
 $\wedge m2av2.type = "2av"$
 $\wedge m2av2.lr = lrn$
 $\wedge m2av2.acc = S$

$$\begin{array}{l}
\wedge m2av2.bal = bal \\
\wedge m2av2.val = val \\
\text{BY } \langle 5 \rangle 7, \langle 5 \rangle 2 \\
\langle 5 \rangle 12. \wedge m2av2 \in msgs \\
\wedge m2av2.type = \text{"2av"} \\
\wedge m2av2.lr = lrn \\
\wedge m2av2.acc = S \\
\wedge m2av2.bal = bal \\
\wedge m2av2.val = val \\
\text{BY } \langle 5 \rangle 11, \text{SafeAcceptorIsAcceptor} \text{ DEF } ReceivedSpec, TypeOK \\
\langle 5 \rangle 13. [lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \in 2avSent[S] \\
\text{BY } \langle 5 \rangle 12, \text{SafeAcceptorIsAcceptor} \text{ DEF } 2avSentSpec2, Proposed \\
\langle 5 \rangle 14. vote1.val = val \text{ BY } \langle 5 \rangle 10, \langle 5 \rangle 13, \langle 5 \rangle 6, \langle 5 \rangle 0 \text{ DEF } 2avSentSpec3 \\
\langle 5 \rangle 20. \text{QED BY } \langle 5 \rangle 1, \langle 5 \rangle 14 \\
\langle 4 \rangle 3. \text{CASE } vote1 \notin votesSent[A] \wedge vote2 \in votesSent[A] \\
\langle 5 \rangle 0. vote2.lr \in Learner \wedge vote2.val \in Value \text{ BY } \langle 4 \rangle 3 \text{ DEF } TypeOK \\
\langle 5 \rangle 1. vote1 = [lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \text{ BY } \langle 4 \rangle 3 \\
\langle 5 \rangle 2. \text{PICK } Q1 \in ByzQuorum : \\
\wedge [lr \mapsto lrn, q \mapsto Q1] \in TrustLive \\
\wedge \forall aa \in Q1 : \\
\quad \exists m \in \{mm \in received[acc] : \\
\quad \quad \wedge mm.type = \text{"2av"} \\
\quad \quad \wedge mm.lr = lrn \\
\quad \quad \wedge mm.bal = bal\} : \\
\quad \quad \wedge m.val = val \\
\quad \quad \wedge m.acc = aa \\
\text{BY } \langle 5 \rangle 1, \langle 2 \rangle 3 \text{ DEF } Phase2b \\
\langle 5 \rangle 3. \langle lrn, vote2.lr \rangle \in Ent \wedge vote2.bal = bal \text{ BY } \langle 5 \rangle 1 \\
\langle 5 \rangle 4. \text{PICK } m2 \in msgs : \\
\wedge m2.type = \text{"2b"} \\
\wedge m2.lr = vote2.lr \\
\wedge m2.acc = A \\
\wedge m2.bal = bal \\
\wedge m2.val = vote2.val \\
\text{BY } \langle 4 \rangle 3, \langle 5 \rangle 3 \text{ DEF } VotesSentSpec1, VotedFor \\
\langle 5 \rangle 5. \text{PICK } Q2 \in ByzQuorum : \\
\wedge [lr \mapsto vote2.lr, q \mapsto Q2] \in TrustLive \\
\wedge \forall ba \in Q2 : \\
\quad \exists m2av \in received[acc] : \\
\quad \quad \wedge m2av.type = \text{"2av"} \\
\quad \quad \wedge m2av.lr = vote2.lr \\
\quad \quad \wedge m2av.acc = ba \\
\quad \quad \wedge m2av.bal = bal \\
\quad \quad \wedge m2av.val = vote2.val \\
\langle 6 \rangle 1. \exists Q2 \in ByzQuorum :
\end{array}$$

$\wedge [lr \mapsto m2.lr, q \mapsto Q2] \in TrustLive$
 $\wedge \forall ba \in Q2 :$
 $\quad \exists m2av \in received[m2.acc] :$
 $\quad \quad \wedge m2av.type = \text{"2av"}$
 $\quad \quad \wedge m2av.lr = m2.lr$
 $\quad \quad \wedge m2av.acc = ba$
 $\quad \quad \wedge m2av.bal = m2.bal$
 $\quad \quad \wedge m2av.val = m2.val$
BY $\langle 5 \rangle 4, \langle 3 \rangle 1$ DEF *MsgInv2b, TypeOK*
 $\langle 6 \rangle 2$. QED BY $\langle 5 \rangle 4, \langle 6 \rangle 1, \langle 3 \rangle 1$
 $\langle 5 \rangle 6$. $\langle lrn, vote2.lr \rangle \in Ent$ BY $\langle 5 \rangle 3$
 $\langle 5 \rangle 7$. PICK $S \in SafeAcceptor : S \in Q1 \wedge S \in Q2$ BY $\langle 5 \rangle 2, \langle 5 \rangle 5, \langle 5 \rangle 6, \langle 5 \rangle 0$, *EntanglementTrustLive*
 $\langle 5 \rangle 8$. PICK $m2av2 \in received[acc] :$
 $\quad \wedge m2av2.type = \text{"2av"}$
 $\quad \wedge m2av2.lr = vote2.lr$
 $\quad \wedge m2av2.acc = S$
 $\quad \wedge m2av2.bal = bal$
 $\quad \wedge m2av2.val = vote2.val$
BY $\langle 5 \rangle 7, \langle 5 \rangle 5$
 $\langle 5 \rangle 9$. $\wedge m2av2 \in msgs$
 $\quad \wedge m2av2.type = \text{"2av"}$
 $\quad \wedge m2av2.lr = vote2.lr$
 $\quad \wedge m2av2.acc = S$
 $\quad \wedge m2av2.bal = bal$
 $\quad \wedge m2av2.val = vote2.val$
BY $\langle 5 \rangle 8, \langle 5 \rangle 0$, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec, TypeOK*
 $\langle 5 \rangle 10$. $[lr \mapsto vote2.lr, bal \mapsto bal, val \mapsto vote2.val] \in 2avSent[S]$
BY $\langle 5 \rangle 9, \langle 5 \rangle 0$ DEF *2avSentSpec2, Proposed*
 $\langle 5 \rangle 11$. PICK $m2av1 \in received[acc] :$
 $\quad \wedge m2av1.type = \text{"2av"}$
 $\quad \wedge m2av1.lr = lrn$
 $\quad \wedge m2av1.acc = S$
 $\quad \wedge m2av1.bal = bal$
 $\quad \wedge m2av1.val = val$
BY $\langle 5 \rangle 7, \langle 5 \rangle 2$
 $\langle 5 \rangle 12$. $\wedge m2av1 \in msgs$
 $\quad \wedge m2av1.type = \text{"2av"}$
 $\quad \wedge m2av1.lr = lrn$
 $\quad \wedge m2av1.acc = S$
 $\quad \wedge m2av1.bal = bal$
 $\quad \wedge m2av1.val = val$
BY $\langle 5 \rangle 11$, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec, TypeOK*
 $\langle 5 \rangle 13$. $[lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \in 2avSent[S]$
BY $\langle 5 \rangle 12$, *SafeAcceptorIsAcceptor* DEF *2avSentSpec2, Proposed*
 $\langle 5 \rangle 14$. $vote2.val = val$ BY $\langle 5 \rangle 10, \langle 5 \rangle 13, \langle 5 \rangle 6, \langle 5 \rangle 0$ DEF *2avSentSpec3*

$\langle 5 \rangle 20$. QED BY $\langle 5 \rangle 1, \langle 5 \rangle 14$
 $\langle 4 \rangle 4$. CASE $vote1 \notin votesSent[A] \wedge vote2 \notin votesSent[A]$ BY $\langle 4 \rangle 4$
 $\langle 4 \rangle 5$. QED BY $\langle 4 \rangle 1, \langle 4 \rangle 2, \langle 4 \rangle 3, \langle 4 \rangle 4$
 $\langle 3 \rangle 2$. CASE $A \neq acc$ BY $\langle 3 \rangle 2$ DEF *VotesSentSpec4*
 $\langle 3 \rangle 3$. QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2$
 $\langle 2 \rangle 4$. QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3$. CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction, Recv, Next, VotesSentSpec4*
 $\langle 1 \rangle 4$. CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction, Disconnect, Next, VotesSentSpec4*
 $\langle 1 \rangle 5$. CASE *LearnerAction* BY $\langle 1 \rangle 5$ DEF *LearnerAction, LearnerRecv, LearnerDecide, Next, VotesSentSpec4*
 $\langle 1 \rangle 6$. CASE *FakeAcceptorAction* BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction, FakeSend, Send, VotesSentSpec4*
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

LEMMA *2avSentSpec1Invariant* \triangleq *Next* \wedge *2avSentSpec1* \Rightarrow *2avSentSpec1'*
 PROOF
 $\langle 1 \rangle$ SUFFICES ASSUME *Next, 2avSentSpec1*,
 NEW $A \in SafeAcceptor$, NEW $p \in 2avSent'[A]$
 PROVE *Proposed(p.lr, A, p.bal, p.val)'*
 BY DEF *2avSentSpec1*
 $\langle 1 \rangle$ USE DEF *2avSentSpec1*
 $\langle 1 \rangle 1$. CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction, Phase1a, Phase1c, Next, Send*
 $\langle 1 \rangle 2$. CASE *AcceptorSendAction*
 $\langle 2 \rangle$ HIDE DEF *Next*
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,
 NEW $bal \in Ballot$,
 NEW $acc \in SafeAcceptor$,
 NEW $val \in Value$,
 $\vee Phase1b(lrn, bal, acc)$
 $\vee Phase2av(lrn, bal, acc, val)$
 $\vee Phase2b(lrn, bal, acc, val)$
 PROVE *Proposed(p.lr, A, p.bal, p.val)'*
 BY $\langle 1 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 2 \rangle 1$. CASE *Phase1b(lrn, bal, acc)* BY $\langle 2 \rangle 1$ DEF *Phase1b*
 $\langle 2 \rangle 2$. CASE *Phase2av(lrn, bal, acc, val)*
 $\langle 3 \rangle$ SUFFICES ASSUME *Send([type \mapsto "2av", $lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]$),*
 $2avSent' = [2avSent \text{ EXCEPT } ![acc] =$
 $2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}$
 PROVE *Proposed(p.lr, A, p.bal, p.val)'*
 BY $\langle 2 \rangle 2$ DEF *Phase2av*
 $\langle 3 \rangle 2$. CASE $acc = A$
 $\langle 4 \rangle 1$. USE DEF *Proposed*
 $\langle 4 \rangle 2$. CASE $p \in 2avSent[acc]$ BY $\langle 3 \rangle 2, \langle 4 \rangle 2, MsgsMonotone$
 $\langle 4 \rangle 3$. CASE $p \notin 2avSent[acc]$
 $\langle 5 \rangle 1$. DEFINE $m0 \triangleq [type \mapsto "2av", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]$
 $\langle 5 \rangle 2$. $m0 \in msgs'$ BY DEF *Phase2b, Send*
 $\langle 5 \rangle 3$. WITNESS $\langle 5 \rangle 2$

$\langle 5 \rangle 10.$ QED BY $\langle 3 \rangle 2, \langle 4 \rangle 3$
 $\langle 4 \rangle 10.$ QED BY $\langle 4 \rangle 2, \langle 4 \rangle 3$
 $\langle 3 \rangle 3.$ CASE $acc \neq A$ BY $\langle 3 \rangle 3$
 $\langle 3 \rangle 4.$ QED BY $\langle 3 \rangle 2, \langle 3 \rangle 3$
 $\langle 2 \rangle 3.$ CASE $Phase2b(lrn, bal, acc, val)$ BY $\langle 2 \rangle 3$ DEF $Phase2b$
 $\langle 2 \rangle 5.$ QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3.$ CASE $AcceptorReceiveAction$ BY $\langle 1 \rangle 3$ DEF $AcceptorReceiveAction, Recv, Next$
 $\langle 1 \rangle 4.$ CASE $AcceptorDisconnectAction$ BY $\langle 1 \rangle 4$ DEF $AcceptorDisconnectAction, Disconnect, Next$
 $\langle 1 \rangle 5.$ CASE $LearnerAction$ BY $\langle 1 \rangle 5$ DEF $LearnerAction, LearnerRecv, LearnerDecide, Next$
 $\langle 1 \rangle 6.$ CASE $FakeAcceptorAction$ BY $\langle 1 \rangle 6$ DEF $FakeAcceptorAction, FakeSend, Send$
 $\langle 1 \rangle 7.$ QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF $Next$

LEMMA $2avSentSpec2Invariant \triangleq Next \wedge 2avSentSpec2 \Rightarrow 2avSentSpec2'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME $Next, 2avSentSpec2,$
NEW $L \in Learner, NEW A \in SafeAcceptor, NEW B \in Ballot, NEW V \in Value,$
 $Proposed(L, A, B, V)'$
PROVE $([lr \mapsto L, bal \mapsto B, val \mapsto V] \in 2avSent[A])'$
BY DEF $2avSentSpec2$
 $\langle 1 \rangle$ USE DEF $2avSentSpec2$
 $\langle 1 \rangle 1.$ CASE $ProposerAction$ BY $\langle 1 \rangle 1$ DEF $ProposerAction, Phase1a, Phase1c, Next, Send$
 $\langle 1 \rangle 2.$ CASE $AcceptorSendAction$
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner,$
NEW $bal \in Ballot,$
NEW $acc \in SafeAcceptor,$
NEW $val \in Value,$
 $\vee Phase1b(lrn, bal, acc)$
 $\vee Phase2av(lrn, bal, acc, val)$
 $\vee Phase2b(lrn, bal, acc, val)$
PROVE $([lr \mapsto L, bal \mapsto B, val \mapsto V] \in 2avSent[A])'$
BY $\langle 1 \rangle 2$ DEF $AcceptorSendAction$
 $\langle 2 \rangle 1.$ CASE $Phase1b(lrn, bal, acc)$ BY $\langle 2 \rangle 1$ DEF $Phase1b$
 $\langle 2 \rangle 2.$ CASE $Phase2av(lrn, bal, acc, val)$
 $\langle 3 \rangle$ SUFFICES ASSUME $Send([type \mapsto "2av", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]),$
 $2avSent' = [2avSent \text{ EXCEPT } ![acc] =$
 $2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}$
PROVE $([lr \mapsto L, bal \mapsto B, val \mapsto V] \in 2avSent[A])'$
BY $\langle 2 \rangle 2$ DEF $Phase2av$
 $\langle 3 \rangle 1.$ QED OBVIOUS
 $\langle 2 \rangle 3.$ CASE $Phase2b(lrn, bal, acc, val)$ BY $\langle 2 \rangle 3$ DEF $Phase2b$
 $\langle 2 \rangle 5.$ QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3.$ CASE $AcceptorReceiveAction$ BY $\langle 1 \rangle 3$ DEF $AcceptorReceiveAction, Recv, Next$
 $\langle 1 \rangle 4.$ CASE $AcceptorDisconnectAction$ BY $\langle 1 \rangle 4$ DEF $AcceptorDisconnectAction, Disconnect, Next$
 $\langle 1 \rangle 5.$ CASE $LearnerAction$ BY $\langle 1 \rangle 5$ DEF $LearnerAction, LearnerRecv, LearnerDecide, Next$
 $\langle 1 \rangle 6.$ CASE $FakeAcceptorAction$ BY $\langle 1 \rangle 6$ DEF $FakeAcceptorAction, FakeSend, Send$

$\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

LEMMA *2avSentSpec3Invariant* \triangleq *Next* \wedge *ConnectedSpec* \wedge *2avSentSpec3* \Rightarrow *2avSentSpec3'*

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME *Next*, *ConnectedSpec*, *2avSentSpec3*,

NEW *L1* \in *Learner*, NEW *L2* \in *Learner*, NEW *A* \in *SafeAcceptor*, NEW *B* \in *Ballot*,

NEW *V1* \in *Value*, NEW *V2* \in *Value*,

$\langle L1, L2 \rangle \in$ *Ent*,

$[lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in 2avSent'[A]$,

$[lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent'[A]$

PROVE *V1* = *V2*

BY DEF *2avSentSpec3*

$\langle 1 \rangle$ USE DEF *2avSentSpec3*

$\langle 1 \rangle 1$. CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction*, *Phase1a*, *Phase1c*, *Next*, *Send*

$\langle 1 \rangle 2$. CASE *AcceptorSendAction*

$\langle 2 \rangle$ SUFFICES ASSUME NEW *lrn* \in *Learner*,

NEW *bal* \in *Ballot*,

NEW *acc* \in *SafeAcceptor*,

NEW *val* \in *Value*,

\vee *Phase1b*(*lrn*, *bal*, *acc*)

\vee *Phase2av*(*lrn*, *bal*, *acc*, *val*)

\vee *Phase2b*(*lrn*, *bal*, *acc*, *val*)

PROVE *V1* = *V2*

BY $\langle 1 \rangle 2$ DEF *AcceptorSendAction*

$\langle 2 \rangle 1$. CASE *Phase1b*(*lrn*, *bal*, *acc*) BY $\langle 2 \rangle 1$ DEF *Phase1b*

$\langle 2 \rangle 2$. CASE *Phase2av*(*lrn*, *bal*, *acc*, *val*)

$\langle 3 \rangle$ SUFFICES

ASSUME NEW *v* \in *Value*,

$\forall P \in \{p \in 2avSent[acc] : p.bal = bal \wedge \langle p.lr, lrn \rangle \in connected[acc]\} : P.val = v$,

$2avSent' = [2avSent \text{ EXCEPT } ![acc] =$

$2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto v]\}$

PROVE *V1* = *V2*

BY $\langle 2 \rangle 2$ DEF *Phase2av*

$\langle 3 \rangle 1$. CASE *A* = *acc*

$\langle 4 \rangle 1$. CASE $\wedge [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in 2avSent[A]$

$\wedge [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent[A]$

BY $\langle 4 \rangle 1, \langle 3 \rangle 1$

$\langle 4 \rangle 3$. CASE $\wedge [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \notin 2avSent[A]$

$\wedge [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent[A]$

$\langle 5 \rangle 1$. $\langle L2, L1 \rangle \in$ *Ent* BY *EntanglementSym*

$\langle 5 \rangle 2$. QED BY $\langle 4 \rangle 3, \langle 3 \rangle 1, \langle 5 \rangle 1$ DEF *ConnectedSpec*

$\langle 4 \rangle 2$. CASE $\wedge [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in 2avSent[A]$

$\wedge [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \notin 2avSent[A]$

BY $\langle 4 \rangle 2, \langle 3 \rangle 1$ DEF *ConnectedSpec*

$\langle 4 \rangle 4$. CASE $\wedge [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \notin 2avSent[A]$

$\wedge [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \notin 2avSent[A]$
 BY $\langle 4 \rangle 4, \langle 3 \rangle 1$
 $\langle 4 \rangle 5$. QED BY $\langle 4 \rangle 1, \langle 4 \rangle 2, \langle 4 \rangle 3, \langle 4 \rangle 4$
 $\langle 3 \rangle 2$. CASE $A \neq acc$ BY $\langle 3 \rangle 2$
 $\langle 3 \rangle 3$. QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2$
 $\langle 2 \rangle 3$. CASE $Phase2b(lrn, bal, acc, val)$ BY $\langle 2 \rangle 3$ DEF $Phase2b$
 $\langle 2 \rangle 5$. QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3$. CASE $AcceptorReceiveAction$ BY $\langle 1 \rangle 3$ DEF $AcceptorReceiveAction, Recv, Next$
 $\langle 1 \rangle 4$. CASE $AcceptorDisconnectAction$ BY $\langle 1 \rangle 4$ DEF $AcceptorDisconnectAction, Disconnect, Next$
 $\langle 1 \rangle 5$. CASE $LearnerAction$ BY $\langle 1 \rangle 5$ DEF $LearnerAction, LearnerRecv, LearnerDecide, Next$
 $\langle 1 \rangle 6$. CASE $FakeAcceptorAction$ BY $\langle 1 \rangle 6$ DEF $FakeAcceptorAction, FakeSend, Send$
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF $Next$

LEMMA $DecisionSpecInvariant \triangleq TypeOK \wedge Next \wedge DecisionSpec \Rightarrow DecisionSpec'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME $Next, TypeOK, DecisionSpec,$
 NEW $L \in Learner$, NEW $B \in Ballot$, NEW $V \in Value$,
 $V \in decision'[L, B]$
 PROVE $ChosenIn(L, B, V)'$
 BY DEF $DecisionSpec$
 $\langle 1 \rangle$ USE DEF $DecisionSpec$
 $\langle 1 \rangle 1$. CASE $ProposerAction$ BY $\langle 1 \rangle 1$ DEF $ProposerAction, Phase1a, Phase1c, Next, Send$
 $\langle 1 \rangle 2$. CASE $AcceptorSendAction$ BY $\langle 1 \rangle 2$ DEF $AcceptorSendAction, Phase1b, Phase2av, Phase2b, Next, Send$
 $\langle 1 \rangle 3$. CASE $AcceptorReceiveAction$ BY $\langle 1 \rangle 3$ DEF $AcceptorReceiveAction, Recv, Next$
 $\langle 1 \rangle 4$. CASE $AcceptorDisconnectAction$ BY $\langle 1 \rangle 4$ DEF $AcceptorDisconnectAction, Disconnect, Next$
 $\langle 1 \rangle 5$. CASE $LearnerAction$
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$,
 $\vee LearnerDecide(lrn, bal)$
 $\vee LearnerRecv(lrn)$
 PROVE $ChosenIn(L, B, V)'$
 BY $\langle 1 \rangle 5$ DEF $LearnerAction$
 $\langle 2 \rangle 2$. CASE $LearnerDecide(lrn, bal)$
 $\langle 3 \rangle 0a$. $TypeOK$ OBVIOUS
 $\langle 3 \rangle 0b$. $TypeOK'$ BY $TypeOKInvariant$
 $\langle 3 \rangle 1$. CASE $V \in decision[L, B]$ BY $\langle 3 \rangle 1, \langle 2 \rangle 2$ DEF $ChosenIn, LearnerDecide$
 $\langle 3 \rangle 2$. CASE $V \notin decision[L, B]$ BY $\langle 3 \rangle 2, \langle 2 \rangle 2, \langle 3 \rangle 0a, \langle 3 \rangle 0b$ DEF $ChosenIn, LearnerDecide, TypeOK$
 $\langle 3 \rangle 3$. QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2$
 $\langle 2 \rangle 3$. CASE $LearnerRecv(lrn)$
 $\langle 3 \rangle 1$. QED BY $\langle 2 \rangle 3$ DEF $LearnerRecv$
 $\langle 2 \rangle 4$. QED BY $\langle 2 \rangle 2, \langle 2 \rangle 3$ DEF $LearnerAction$
 $\langle 1 \rangle 6$. CASE $FakeAcceptorAction$ BY $\langle 1 \rangle 6$ DEF $FakeAcceptorAction, FakeSend, Send$
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF $Next$

LEMMA $ConnectedSpecInvariant \triangleq Next \wedge ConnectedSpec \Rightarrow ConnectedSpec'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME $Next$, $ConnectedSpec$,
NEW $A \in SafeAcceptor$,
NEW $L1 \in Learner$, NEW $L2 \in Learner$,
 $\langle L1, L2 \rangle \in Ent$
PROVE $\langle L1, L2 \rangle \in connected'[A]$
BY DEF $ConnectedSpec$
 $\langle 1 \rangle$ USE DEF $ConnectedSpec$
 $\langle 1 \rangle 1$.CASE $ProposerAction$ BY $\langle 1 \rangle 1$ DEF $ProposerAction$, $Phase1a$, $Phase1c$, $Next$
 $\langle 1 \rangle 2$.CASE $AcceptorSendAction$ BY $\langle 1 \rangle 2$ DEF $AcceptorSendAction$, $Phase1b$, $Phase2b$, $Phase2av$, $Next$
 $\langle 1 \rangle 3$.CASE $AcceptorReceiveAction$ BY $\langle 1 \rangle 3$ DEF $AcceptorReceiveAction$, $Recv$, $Next$
 $\langle 1 \rangle 4$.CASE $AcceptorDisconnectAction$ BY $\langle 1 \rangle 4$ DEF $AcceptorDisconnectAction$, $Disconnect$, $Next$
 $\langle 1 \rangle 5$.CASE $LearnerAction$ BY $\langle 1 \rangle 5$ DEF $LearnerAction$, $LearnerRecv$, $LearnerDecide$, $Next$
 $\langle 1 \rangle 6$.CASE $FakeAcceptorAction$ BY $\langle 1 \rangle 6$ DEF $FakeAcceptorAction$, $FakeSend$, $Send$
 $\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$, $\langle 1 \rangle 4$, $\langle 1 \rangle 5$, $\langle 1 \rangle 6$ DEF $Next$

LEMMA $MsgInvInvariant \triangleq$

$TypeOK \wedge MsgInv \wedge VotesSentSpec1 \wedge VotesSentSpec2 \wedge VotesSentSpec3 \wedge 2avSentSpec1 \wedge$
 $Next \Rightarrow MsgInv'$

PROOF

$\langle 1 \rangle$ USE DEF $MsgInv$
 $\langle 1 \rangle 1b$. ASSUME $TypeOK$, $VotesSentSpec1$, $VotesSentSpec2$, $VotesSentSpec3$, $2avSentSpec1$, $Next$,
 $\forall m \in msgs : m.acc \in SafeAcceptor \wedge m.type = "1b" \Rightarrow MsgInv1b(m)$,
NEW $m \in msgs'$, $m.acc \in SafeAcceptor$, $m.type = "1b"$
PROVE $MsgInv1b(m)'$
 $\langle 2 \rangle 0$. $TypeOK$ BY $\langle 1 \rangle 1b$
 $\langle 2 \rangle 0a$. $TypeOK'$ BY $\langle 1 \rangle 1b$, $TypeOKInvariant$
 $\langle 2 \rangle 0b$. $m \in Message$ BY $\langle 2 \rangle 0a$ DEF $TypeOK$
 $\langle 2 \rangle 0c$. $maxBal \in [Learner \times Acceptor \rightarrow Ballot]$ BY $\langle 1 \rangle 1b$ DEF $TypeOK$
 $\langle 2 \rangle 0d$. $maxBal' \in [Learner \times Acceptor \rightarrow Ballot]$ BY $\langle 2 \rangle 0a$ DEF $TypeOK$
 $\langle 2 \rangle 0e$. $m.type = "1b"$ BY $\langle 1 \rangle 1b$
 $\langle 2 \rangle 0f$. $m.bal \in Ballot$ BY $\langle 2 \rangle 0b$, $\langle 2 \rangle 0e$ DEF $Message$, $Ballot$
 $\langle 2 \rangle 0g$. $maxBal[m.lr, m.acc] \in Ballot$ BY $\langle 2 \rangle 0b$, $\langle 2 \rangle 0c$, $\langle 2 \rangle 0e$ DEF $Message$
 $\langle 2 \rangle 0h$. $maxBal'[m.lr, m.acc] \in Ballot$ BY $\langle 2 \rangle 0b$, $\langle 2 \rangle 0d$, $\langle 2 \rangle 0e$ DEF $Message$
 $\langle 2 \rangle 0i$. $maxBal[m.lr, m.acc] \leq maxBal'[m.lr, m.acc]$ BY $\langle 1 \rangle 1b$, $\langle 2 \rangle 0b$, $MaxBalMonotone$ DEF $TypeOK$, $Message$
 $\langle 2 \rangle 1$.CASE $ProposerAction$
 $\langle 3 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $proposer \in Ballot$, NEW $val \in Value$,
 $\vee Phase1a(lrn, proposer)$
 $\vee Phase1c(lrn, proposer, val)$
PROVE $MsgInv1b(m)'$
BY $\langle 2 \rangle 1$, $ValueNotEmpty$ DEF $ProposerAction$
 $\langle 3 \rangle 1$.CASE $Phase1a(lrn, proposer)$
 $\langle 4 \rangle 1$. $m \in msgs$ BY $\langle 3 \rangle 1$, $\langle 2 \rangle 0e$ DEF $Phase1a$, $Send$
 $\langle 4 \rangle 2$. QED BY $\langle 1 \rangle 1b$, $\langle 4 \rangle 1$, $\langle 3 \rangle 1$ DEF $Phase1a$, $MsgInv1b$
 $\langle 3 \rangle 2$.CASE $Phase1c(lrn, proposer, val)$
 $\langle 4 \rangle 1$. $m \in msgs$ BY $\langle 3 \rangle 2$, $\langle 2 \rangle 0e$ DEF $Phase1c$, $Send$, $TypeOK$

$\langle 4 \rangle 2.$ QED BY $\langle 1 \rangle 1b, \langle 4 \rangle 1, \langle 3 \rangle 2$ DEF *Phase1c, MsgInv1b*
 $\langle 3 \rangle 3.$ QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2$
 $\langle 2 \rangle 2.$ CASE *AcceptorSendAction*
 $\langle 3 \rangle$ SUFFICES ASSUME NEW $lrn \in \text{Learner},$
NEW $bal \in \text{Ballot},$
NEW $acc \in \text{SafeAcceptor},$
NEW $val \in \text{Value},$
 $\vee \text{Phase1b}(lrn, bal, acc)$
 $\vee \text{Phase2av}(lrn, bal, acc, val)$
 $\vee \text{Phase2b}(lrn, bal, acc, val)$
PROVE $\text{MsgInv1b}(m)'$
BY $\langle 2 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 3 \rangle 1.$ CASE *Phase1b*(lrn, bal, acc)
 $\langle 4 \rangle 1.$ $m.bal \leq \text{maxBal}'[m.lr, m.acc]$
 $\langle 5 \rangle 6.$ CASE $m \in \text{msgs}$
 $\langle 6 \rangle 0.$ $m.bal \leq \text{maxBal}[m.lr, m.acc]$ BY $\langle 1 \rangle 1b, \langle 5 \rangle 6$ DEF *MsgInv1b*
 $\langle 6 \rangle 1.$ QED BY $\langle 6 \rangle 0, \langle 2 \rangle 0i, \langle 2 \rangle 0g, \langle 2 \rangle 0h, \langle 2 \rangle 0b, \text{BallotLeqTrans}$ DEF *Message*
 $\langle 5 \rangle 7.$ CASE $m \notin \text{msgs}$
 $\langle 6 \rangle 0.$ $m = [type \mapsto "1b", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal,$
 $votes \mapsto \{p \in \text{votesSent}[acc] : \text{MaxVote}(acc, bal, p)\},$
 $proposals \mapsto \{p \in \text{2avSent}[acc] : p.bal < bal \wedge p.lr = lrn\}]$
BY $\langle 3 \rangle 1, \langle 5 \rangle 7$ DEF *Next, Phase1b, Send*
 $\langle 6 \rangle 3.$ SUFFICES $bal \leq \text{maxBal}'[lrn, acc]$ BY $\langle 6 \rangle 0$
 $\langle 6 \rangle 4.$ $\text{maxBal}' = [\text{maxBal} \text{ EXCEPT } ![lrn, acc] = bal]$ BY $\langle 3 \rangle 1$ DEF *Phase1b, Send*
 $\langle 6 \rangle 5.$ $\text{maxBal}'[(lrn, acc)] = bal$ BY $\langle 6 \rangle 4, \langle 2 \rangle 0c, \langle 2 \rangle 0d$
 $\langle 6 \rangle 6.$ QED BY $\langle 6 \rangle 0, \langle 6 \rangle 5$ DEF *Ballot*
 $\langle 5 \rangle 8.$ QED BY $\langle 5 \rangle 6, \langle 5 \rangle 7$
 $\langle 4 \rangle 5.$ $(m.votes = \{p \in \text{votesSent}[m.acc] : \text{MaxVote}(m.acc, m.bal, p)\})'$
 $\langle 5 \rangle 1.$ CASE $m \in \text{msgs}$ BY $\langle 1 \rangle 1b, \langle 3 \rangle 1, \langle 5 \rangle 1$ DEF *MsgInv1b, Phase1b*
 $\langle 5 \rangle 2.$ CASE $m \notin \text{msgs}$
 $\langle 6 \rangle 0.$ $m = [type \mapsto "1b", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal,$
 $votes \mapsto \{p \in \text{votesSent}[acc] : \text{MaxVote}(acc, bal, p)\},$
 $proposals \mapsto \{p \in \text{2avSent}[acc] : p.bal < bal \wedge p.lr = lrn\}]$
BY $\langle 3 \rangle 1, \langle 5 \rangle 2$ DEF *Phase1b, Send*
 $\langle 6 \rangle 2.$ QED BY $\langle 6 \rangle 0, \langle 3 \rangle 1$ DEF *Phase1b, Send*
 $\langle 5 \rangle 3.$ QED BY $\langle 5 \rangle 1, \langle 5 \rangle 2$
 $\langle 4 \rangle 6.$ $(m.proposals = \{p \in \text{2avSent}[m.acc] : p.bal < m.bal \wedge p.lr = m.lr\})'$
 $\langle 5 \rangle 1.$ CASE $m \in \text{msgs}$ BY $\langle 1 \rangle 1b, \langle 3 \rangle 1, \langle 5 \rangle 1$ DEF *Phase1b, MsgInv1b*
 $\langle 5 \rangle 2.$ CASE $m \notin \text{msgs}$
 $\langle 6 \rangle 0.$ $m = [type \mapsto "1b", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal,$
 $votes \mapsto \{p \in \text{votesSent}[acc] : \text{MaxVote}(acc, bal, p)\},$
 $proposals \mapsto \{p \in \text{2avSent}[acc] :$
 $p.bal < bal \wedge p.lr = lrn\}]$ BY $\langle 3 \rangle 1, \langle 5 \rangle 2$ DEF *Phase1b, Send*
 $\langle 6 \rangle 2.$ QED BY $\langle 6 \rangle 0, \langle 3 \rangle 1$ DEF *Phase1b, Send*
 $\langle 5 \rangle 3.$ QED BY $\langle 5 \rangle 1, \langle 5 \rangle 2$

$\langle 4 \rangle 10.$ QED BY $\langle 4 \rangle 1, \langle 4 \rangle 5, \langle 4 \rangle 6$ DEF *MsgInv1b*
 $\langle 3 \rangle 2.$ CASE *Phase2av*(*lrn*, *bal*, *acc*, *val*)
 $\langle 4 \rangle$ SUFFICES
 ASSUME $\text{maxBal}[\text{lrn}, \text{acc}] \leq \text{bal}$,
 $\text{Send}([\text{type} \mapsto \text{"2av"}, \text{lr} \mapsto \text{lrn}, \text{acc} \mapsto \text{acc}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]),$
 $2\text{avSent}' = [2\text{avSent} \text{ EXCEPT } ![\text{acc}] =$
 $2\text{avSent}[\text{acc}] \cup \{[\text{lr} \mapsto \text{lrn}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]\}$
 PROVE *MsgInv1b*(*m*)'
 BY $\langle 3 \rangle 2$ DEF *Phase2av*
 $\langle 4 \rangle 1.$ $m \in \text{msgs}$ BY $\langle 2 \rangle 0e$ DEF *Send*
 $\langle 4 \rangle 1a.$ $m.\text{acc} \in \text{Acceptor}$ BY $\langle 4 \rangle 1, \text{MessageType}, \langle 2 \rangle 0e, \langle 2 \rangle 0$ DEF *TypeOK*
 $\langle 4 \rangle 2.$ $(m.\text{bal} \leq \text{maxBal}[m.\text{lr}, m.\text{acc}])'$ BY $\langle 1 \rangle 1b, \langle 4 \rangle 1, \langle 3 \rangle 2$ DEF *Phase2av, Send, MsgInv1b*
 $\langle 4 \rangle 4.$ $(m.\text{votes} = \{p \in \text{votesSent}[m.\text{acc}] : \text{MaxVote}(m.\text{acc}, m.\text{bal}, p)\})'$
 BY $\langle 1 \rangle 1b, \langle 4 \rangle 1, \langle 3 \rangle 2$ DEF *Phase2av, Send, MsgInv1b*
 $\langle 4 \rangle 5.$ $(m.\text{proposals} = \{p \in 2\text{avSent}[m.\text{acc}] : p.\text{bal} < m.\text{bal} \wedge p.\text{lr} = m.\text{lr}\})'$
 $\langle 5 \rangle 1.$ CASE $m.\text{acc} \neq \text{acc}$
 $\langle 6 \rangle 1.$ $2\text{avSent}'[m.\text{acc}] = 2\text{avSent}[m.\text{acc}]$ BY $\langle 3 \rangle 2, \langle 4 \rangle 1, \langle 5 \rangle 1, \langle 2 \rangle 0, \langle 2 \rangle 0e, \text{MessageType}$ DEF *Phase2b*,
 $\langle 6 \rangle 2.$ QED BY $\langle 6 \rangle 1, \langle 4 \rangle 1, \langle 1 \rangle 1b, \langle 2 \rangle 0e$ DEF *MsgInv1b*
 $\langle 5 \rangle 2.$ CASE $m.\text{acc} = \text{acc}$
 $\langle 6 \rangle 1.$ $m.\text{bal} \leq \text{maxBal}[m.\text{lr}, m.\text{acc}]$ BY $\langle 1 \rangle 1b, \langle 4 \rangle 1$ DEF *MsgInv1b*
 $\langle 6 \rangle 3.$ $m.\text{bal} \in \text{Ballot}$ BY $\langle 2 \rangle 0a, \text{MessageType}$ DEF *TypeOK*
 $\langle 6 \rangle 5.$ SUFFICES $\{p \in 2\text{avSent}[\text{acc}] : p.\text{bal} < m.\text{bal} \wedge p.\text{lr} = m.\text{lr}\} =$
 $\{p \in 2\text{avSent}'[\text{acc}] : p.\text{bal} < m.\text{bal} \wedge p.\text{lr} = m.\text{lr}\}$
 BY $\langle 4 \rangle 1, \langle 2 \rangle 0e, \langle 1 \rangle 1b, \langle 5 \rangle 2$ DEF *MsgInv1b*
 $\langle 6 \rangle 6.$ SUFFICES ASSUME NEW $p \in 2\text{avSent}'[\text{acc}], p.\text{bal} < m.\text{bal}, p.\text{lr} = m.\text{lr}$
 PROVE $p \in 2\text{avSent}[\text{acc}]$ BY $\langle 2 \rangle 0a, \text{SafeAcceptorIsAcceptor}$ DEF *TypeOK*
 $\langle 6 \rangle 7.$ CASE $p \in 2\text{avSent}[\text{acc}]$ BY $\langle 6 \rangle 7$
 $\langle 6 \rangle 8.$ CASE $p \notin 2\text{avSent}[\text{acc}]$
 $\langle 7 \rangle 1.$ $p = [\text{lr} \mapsto \text{lrn}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]$ BY $\langle 6 \rangle 8, \langle 2 \rangle 0a, \text{SafeAcceptorIsAcceptor}$ DEF *TypeOK*
 $\langle 7 \rangle 2.$ $\text{maxBal}[m.\text{lr}, m.\text{acc}] \leq \text{bal}$ BY $\langle 5 \rangle 2, \langle 7 \rangle 1, \langle 6 \rangle 6$
 $\langle 7 \rangle 4.$ $m.\text{bal} \leq \text{bal}$ BY $\langle 6 \rangle 1, \langle 7 \rangle 2, \langle 6 \rangle 3, \langle 2 \rangle 0g, \text{BallotLeqTrans}$
 $\langle 7 \rangle 10.$ QED BY $\langle 7 \rangle 1, \langle 7 \rangle 4, \langle 6 \rangle 6, \langle 6 \rangle 3, \text{BallotLeNotLeq}$
 $\langle 6 \rangle 10.$ QED BY $\langle 6 \rangle 7, \langle 6 \rangle 8$
 $\langle 5 \rangle 3.$ QED BY $\langle 5 \rangle 1, \langle 5 \rangle 2$
 $\langle 4 \rangle 10.$ QED BY $\langle 4 \rangle 2, \langle 4 \rangle 4, \langle 4 \rangle 5$ DEF *MsgInv1b*
 $\langle 3 \rangle 3.$ CASE *Phase2b*(*lrn*, *bal*, *acc*, *val*)
 $\langle 4 \rangle$ SUFFICES
 ASSUME $\forall L \in \text{Learner} : \text{maxBal}[L, \text{acc}] \leq \text{bal}$,
 $\text{Send}([\text{type} \mapsto \text{"2b"}, \text{lr} \mapsto \text{lrn}, \text{acc} \mapsto \text{acc}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]),$
 $\text{votesSent}' = [\text{votesSent} \text{ EXCEPT }$
 $![\text{acc}] = \text{votesSent}[\text{acc}] \cup \{[\text{lr} \mapsto \text{lrn}, \text{bal} \mapsto \text{bal}, \text{val} \mapsto \text{val}]\}$
 PROVE *MsgInv1b*(*m*)'
 BY $\langle 3 \rangle 3$ DEF *Phase2b*
 $\langle 4 \rangle 1.$ $m \in \text{msgs}$ BY $\langle 2 \rangle 0e$ DEF *Send*
 $\langle 4 \rangle 1a.$ $m.\text{acc} \in \text{Acceptor}$ BY $\langle 4 \rangle 1, \text{MessageType}, \langle 2 \rangle 0e, \langle 2 \rangle 0$ DEF *TypeOK*

$\langle 4 \rangle 2. (m.bal \leq \max Bal[m.lr, m.acc])'$ BY $\langle 4 \rangle 1, \langle 1 \rangle 1b, \langle 3 \rangle 3$ DEF *Phase2b, MsgInv1b*
 $\langle 4 \rangle 4. (m.proposals = \{p \in 2avSent[m.acc] : p.bal < m.bal \wedge p.lr = m.lr\})'$
BY $\langle 4 \rangle 1, \langle 1 \rangle 1b, \langle 3 \rangle 3$ DEF *Phase2b, MsgInv1b*
 $\langle 4 \rangle 5. (m.votes = \{p \in votesSent[m.acc] : MaxVote(m.acc, m.bal, p)\})'$
 $\langle 5 \rangle 1.$ CASE $m.acc \neq acc$
 $\langle 6 \rangle 1. votesSent'[m.acc] = votesSent[m.acc]$ BY $\langle 3 \rangle 3, \langle 4 \rangle 1, \langle 5 \rangle 1, \langle 2 \rangle 0, \langle 2 \rangle 0e$, *MessageType* DEF *Phase2b*
 $\langle 6 \rangle 2.$ QED BY $\langle 6 \rangle 1, \langle 4 \rangle 1, \langle 1 \rangle 1b, \langle 2 \rangle 0e$ DEF *MsgInv1b*
 $\langle 5 \rangle 2.$ CASE $m.acc = acc$
 $\langle 6 \rangle 1. m.bal \leq \max Bal[m.lr, m.acc]$ BY $\langle 1 \rangle 1b, \langle 4 \rangle 1$ DEF *MsgInv1b*
 $\langle 6 \rangle 2. \max Bal[m.lr, m.acc] \leq bal$ BY $\langle 2 \rangle 0a, \langle 2 \rangle 0e, \langle 5 \rangle 2, MessageType$ DEF *Ballot, TypeOK*
 $\langle 6 \rangle 3. m.bal \in Ballot$ BY $\langle 2 \rangle 0a, MessageType$ DEF *TypeOK*
 $\langle 6 \rangle 4. m.bal \leq bal$ BY $\langle 6 \rangle 1, \langle 6 \rangle 2, \langle 6 \rangle 3, \langle 2 \rangle 0g, BallotLeqTrans$
 $\langle 6 \rangle 5.$ SUFFICES $\{p \in votesSent[acc] : MaxVote(acc, m.bal, p)\} =$
 $\{p \in votesSent'[acc] : MaxVote(acc, m.bal, p)\}'$
BY $\langle 4 \rangle 1, \langle 2 \rangle 0e, \langle 1 \rangle 1b, \langle 5 \rangle 2$ DEF *MsgInv1b*
 $\langle 6 \rangle 6. \{p \in votesSent[acc] : MaxVote(acc, m.bal, p)\} \subseteq$
 $\{p \in votesSent'[acc] : MaxVote(acc, m.bal, p)\}'$
BY $\langle 4 \rangle 1a, \langle 2 \rangle 0, VotesSentMonotone, \langle 6 \rangle 4$ DEF *TypeOK*
 $\langle 6 \rangle 7. \{p \in votesSent'[acc] : MaxVote(acc, m.bal, p)\}' \subseteq$
 $\{p \in votesSent[acc] : MaxVote(acc, m.bal, p)\}$
 $\langle 7 \rangle 1.$ SUFFICES ASSUME NEW $p \in votesSent'[acc],$
 $MaxVote(acc, m.bal, p)',$
 $p \notin votesSent[acc]$
PROVE FALSE
OBVIOUS
 $\langle 7 \rangle 2. p = [lr \mapsto lrn, bal \mapsto bal, val \mapsto val]$ BY $\langle 7 \rangle 1, \langle 5 \rangle 2, \langle 2 \rangle 0$ DEF *TypeOK*
 $\langle 7 \rangle 3.$ QED BY $\langle 7 \rangle 2, \langle 7 \rangle 1, \langle 6 \rangle 4, \langle 6 \rangle 3, BallotLeNotLeq$ DEF *MaxVote*
 $\langle 6 \rangle 8.$ QED BY $\langle 6 \rangle 6, \langle 6 \rangle 7$
 $\langle 5 \rangle 3.$ QED BY $\langle 5 \rangle 1, \langle 5 \rangle 2$
 $\langle 4 \rangle 6.$ QED BY $\langle 4 \rangle 2, \langle 4 \rangle 4, \langle 4 \rangle 5$ DEF *MsgInv1b*
 $\langle 3 \rangle 4.$ QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2, \langle 3 \rangle 3$
 $\langle 2 \rangle 4.$ CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 1b, \langle 2 \rangle 4$ DEF *AcceptorReceiveAction, Recv, MsgInv1b, Next*
 $\langle 2 \rangle 5.$ CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 1b, \langle 2 \rangle 5$ DEF *AcceptorDisconnectAction, Disconnect, MsgInv1b, Next*
 $\langle 2 \rangle 6.$ CASE *LearnerAction* BY $\langle 1 \rangle 1b, \langle 2 \rangle 6$ DEF *LearnerAction, LearnerRecv, LearnerDecide, MsgInv1b, Next*
 $\langle 2 \rangle 7.$ CASE *FakeAcceptorAction* BY $\langle 1 \rangle 1b, \langle 2 \rangle 7, SafeAcceptorAssumption$ DEF *FakeAcceptorAction, FakeSend, Next*
 $\langle 2 \rangle 8.$ QED BY $\langle 1 \rangle 1b, \langle 2 \rangle 0a, \langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 4, \langle 2 \rangle 5, \langle 2 \rangle 6, \langle 2 \rangle 7$ DEF *Next*
 $\langle 1 \rangle 2av.$ ASSUME *TypeOK, Next,*
 $\forall m \in msgs : m.acc \in SafeAcceptor \wedge m.type = "2av" \Rightarrow MsgInv2av(m),$
NEW $m \in msgs', m.acc \in SafeAcceptor, m.type = "2av"$
PROVE *MsgInv2av(m)'*
 $\langle 2 \rangle 0a.$ *TypeOK* BY $\langle 1 \rangle 2av$
 $\langle 2 \rangle 0b.$ *TypeOK'* BY $\langle 1 \rangle 2av, TypeOKInvariant$
 $\langle 2 \rangle 0e.$ $m.type = "2av"$ BY $\langle 1 \rangle 2av$
 $\langle 2 \rangle 1.$ CASE *ProposerAction*
 $\langle 3 \rangle 0.$ $m \in msgs$ BY $\langle 1 \rangle 2av, \langle 2 \rangle 1, \langle 2 \rangle 0e$ DEF *ProposerAction, Phase1a, Phase1c, MsgInv2av, Next, Send*

$\langle 3 \rangle 1.$ QED BY $\langle 1 \rangle 2av, \langle 3 \rangle 0, \langle 2 \rangle 1, \langle 2 \rangle 0e$ DEF *ProposerAction, Phase1a, Phase1c, MsgInv2av, Next, Send*
 $\langle 2 \rangle 2.$ CASE *AcceptorSendAction*
 $\langle 3 \rangle$ SUFFICES ASSUME NEW $lrn \in \text{Learner},$
NEW $bal \in \text{Ballot},$
NEW $acc \in \text{SafeAcceptor},$
NEW $val \in \text{Value},$
 $\vee \text{Phase1b}(lrn, bal, acc)$
 $\vee \text{Phase2av}(lrn, bal, acc, val)$
 $\vee \text{Phase2b}(lrn, bal, acc, val)$
PROVE $\text{MsgInv2av}(m)'$
BY $\langle 2 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 3 \rangle 1.$ CASE *Phase1b*(lrn, bal, acc)
 $\langle 4 \rangle 1.$ $m \in \text{msgs}$ BY $\langle 3 \rangle 1, \langle 2 \rangle 0e$ DEF *Phase1b, Send*
 $\langle 4 \rangle 2.$ QED BY $\langle 1 \rangle 2av, \langle 4 \rangle 1, \langle 3 \rangle 1$ DEF *Phase1b, MsgInv2av, Send*
 $\langle 3 \rangle 2.$ CASE *Phase2av*(lrn, bal, acc, val)
 $\langle 4 \rangle$ SUFFICES
ASSUME $\text{InitializedBallot}(lrn, bal),$
 $\text{AnnouncedValue}(lrn, bal, val),$
 $\text{KnowsSafeAt}(lrn, acc, bal, val),$
 $\text{Send}([type \mapsto "2av", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]),$
 $2avSent' = [2avSent \text{ EXCEPT } ![acc] =$
 $2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\},$
UNCHANGED *received*
PROVE $\text{MsgInv2av}(m)'$
BY $\langle 3 \rangle 2$ DEF *Phase2av*
 $\langle 4 \rangle 1.$ CASE $m \in \text{msgs}$
 $\langle 5 \rangle 1.$ $\text{InitializedBallot}(m.lr, m.bal)'$ BY $\langle 4 \rangle 1, \langle 2 \rangle 0e, \langle 1 \rangle 2av, \text{MsgsMonotone}$ DEF *MsgInv2av, InitializedBallot*
 $\langle 5 \rangle 2.$ $\text{AnnouncedValue}(m.lr, m.bal, m.val)'$ BY $\langle 4 \rangle 1, \langle 2 \rangle 0e, \langle 1 \rangle 2av, \text{MsgsMonotone}$ DEF *MsgInv2av, AnnouncedValue*
 $\langle 5 \rangle 3.$ $\text{KnowsSafeAt}(m.lr, m.acc, m.bal, m.val)'$ BY $\langle 4 \rangle 1, \langle 1 \rangle 2av$ DEF *Phase2av, MsgInv2av*
 $\langle 5 \rangle 4.$ $[lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in 2avSent'[m.acc]$
BY $\langle 4 \rangle 1, \langle 2 \rangle 0e, \langle 1 \rangle 2av, 2avSentMonotone, MessageType$ DEF *MsgInv2av, TypeOK*
 $\langle 5 \rangle 5.$ $(\exists Q \in \text{ByzQuorum} :$
 $\wedge [lr \mapsto m.lr, q \mapsto Q] \in \text{TrustLive}$
 $\wedge \forall ba \in Q :$
 $\exists m1b \in \text{received}[m.acc] :$
 $\wedge m1b.type = "1b"$
 $\wedge m1b.lr = m.lr$
 $\wedge m1b.acc = ba$
 $\wedge m1b.bal = m.bal)'$
BY $\langle 4 \rangle 1, \langle 2 \rangle 0e, \langle 1 \rangle 2av, 2avSentMonotone, MessageType$ DEF *MsgInv2av, TypeOK*
 $\langle 5 \rangle 6.$ QED BY $\langle 5 \rangle 1, \langle 5 \rangle 2, \langle 5 \rangle 3, \langle 5 \rangle 4, \langle 5 \rangle 5$ DEF *MsgInv2av*
 $\langle 4 \rangle 2.$ CASE $m \notin \text{msgs}$
 $\langle 5 \rangle 1.$ $m = [type \mapsto "2av", lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]$ BY $\langle 4 \rangle 2$ DEF *Send*
 $\langle 5 \rangle 3.$ $\text{InitializedBallot}(m.lr, m.bal)'$ BY $\langle 5 \rangle 1, \langle 3 \rangle 2$ DEF *Phase2av*
 $\langle 5 \rangle 4.$ $\text{AnnouncedValue}(m.lr, m.bal, m.val)'$ BY $\langle 5 \rangle 1$

$\langle 5 \rangle 5. \text{KnowsSafeAt}(m.lr, m.acc, m.bal, m.val)' \text{BY } \langle 5 \rangle 1$
 $\langle 5 \rangle 6. ([lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in 2avSent[m.acc])' \text{BY } \langle 5 \rangle 1, \langle 2 \rangle 0b \text{ DEF } TypeOK$
 $\langle 5 \rangle 7. (\exists Q \in ByzQuorum :$
 $\quad \wedge [lr \mapsto m.lr, q \mapsto Q] \in TrustLive$
 $\quad \wedge \forall ba \in Q :$
 $\quad \quad \exists m1b \in received[m.acc] :$
 $\quad \quad \quad \wedge m1b.type = "1b"$
 $\quad \quad \quad \wedge m1b.lr = m.lr$
 $\quad \quad \quad \wedge m1b.acc = ba$
 $\quad \quad \quad \wedge m1b.bal = m.bal)'$
 $\langle 6 \rangle 1. \text{CASE } KnowsSafeAt1(lrn, acc, bal, val)$
 $\quad \langle 7 \rangle 1. \text{PICK } Q1 \in ByzQuorum :$
 $\quad \quad \wedge [lr \mapsto lrn, q \mapsto Q1] \in TrustLive$
 $\quad \quad \wedge \forall a \in Q1 :$
 $\quad \quad \quad \exists m1b \in received[acc] :$
 $\quad \quad \quad \quad \wedge m1b.type = "1b"$
 $\quad \quad \quad \quad \wedge m1b.lr = lrn$
 $\quad \quad \quad \quad \wedge m1b.bal = bal$
 $\quad \quad \quad \quad \wedge m1b.acc = a$
 $\quad \text{BY } \langle 6 \rangle 1 \text{ DEF } KnowsSafeAt1$
 $\quad \langle 7 \rangle 2. \text{WITNESS } Q1 \in ByzQuorum$
 $\quad \langle 7 \rangle 3. \text{QED BY } \langle 7 \rangle 1, \langle 5 \rangle 1$
 $\langle 6 \rangle 2. \text{CASE } KnowsSafeAt2(lrn, acc, bal, val)$
 $\quad \langle 7 \rangle 1. \text{PICK } Q2 \in ByzQuorum :$
 $\quad \quad \wedge [lr \mapsto lrn, q \mapsto Q2] \in TrustLive$
 $\quad \quad \wedge \forall a \in Q2 :$
 $\quad \quad \quad \exists m1b \in received[acc] :$
 $\quad \quad \quad \quad \wedge m1b.type = "1b"$
 $\quad \quad \quad \quad \wedge m1b.lr = lrn$
 $\quad \quad \quad \quad \wedge m1b.bal = bal$
 $\quad \quad \quad \quad \wedge m1b.acc = a$
 $\quad \text{BY } \langle 6 \rangle 2 \text{ DEF } KnowsSafeAt2$
 $\quad \langle 7 \rangle 2. \text{WITNESS } Q2 \in ByzQuorum$
 $\quad \langle 7 \rangle 3. \text{QED BY } \langle 7 \rangle 1, \langle 5 \rangle 1$
 $\quad \langle 6 \rangle 3. \text{QED BY } \langle 6 \rangle 1, \langle 6 \rangle 2 \text{ DEF } KnowsSafeAt$
 $\langle 5 \rangle 8. \text{QED BY } \langle 5 \rangle 1, \langle 5 \rangle 3, \langle 5 \rangle 4, \langle 5 \rangle 5, \langle 5 \rangle 6, \langle 5 \rangle 7, Message\ Type \text{ DEF } MsgInv2av, TypeOK$
 $\langle 4 \rangle 20. \text{QED BY } \langle 4 \rangle 1, \langle 4 \rangle 2$
 $\langle 3 \rangle 3. \text{CASE } Phase2b(lrn, bal, acc, val)$
 $\quad \langle 4 \rangle 1. m \in msgs \text{BY } \langle 3 \rangle 3, \langle 2 \rangle 0e \text{ DEF } Phase2b, Send$
 $\quad \langle 4 \rangle 2. \text{QED BY } \langle 1 \rangle 2av, \langle 4 \rangle 1, \langle 3 \rangle 3 \text{ DEF } Phase2b, MsgInv2av, Send$
 $\quad \langle 3 \rangle 4. \text{QED BY } \langle 3 \rangle 1, \langle 3 \rangle 2, \langle 3 \rangle 3$
 $\langle 2 \rangle 4. \text{CASE } AcceptorReceiveAction$
 $\quad \langle 3 \rangle 1. m \in msgs \text{BY } \langle 2 \rangle 4 \text{ DEF } AcceptorReceiveAction, Recv$
 $\quad \langle 3 \rangle 6. (\exists Q \in ByzQuorum :$
 $\quad \quad \wedge [lr \mapsto m.lr, q \mapsto Q] \in TrustLive$

$$\begin{aligned}
& \wedge \forall ba \in Q : \\
& \quad \exists m1b \in received[m.acc] : \\
& \quad \quad \wedge m1b.type = \text{"1b"} \\
& \quad \quad \wedge m1b.lr = m.lr \\
& \quad \quad \wedge m1b.acc = ba \\
& \quad \quad \wedge m1b.bal = m.bal)' \\
\langle 7 \rangle 1. \text{ PICK } Q0 \in ByzQuorum : \\
& \quad \wedge [lr \mapsto m.lr, q \mapsto Q0] \in TrustLive \\
& \quad \wedge \forall ba \in Q0 : \\
& \quad \quad \exists m1b \in received[m.acc] : \\
& \quad \quad \quad \wedge m1b.type = \text{"1b"} \\
& \quad \quad \quad \wedge m1b.lr = m.lr \\
& \quad \quad \quad \wedge m1b.acc = ba \\
& \quad \quad \quad \wedge m1b.bal = m.bal \\
& \quad \text{BY } \langle 1 \rangle 2av, \langle 3 \rangle 1, \langle 2 \rangle 0e \text{ DEF } MsgInv2av \\
\langle 7 \rangle 2. \text{ WITNESS } Q0 \in ByzQuorum \\
\langle 7 \rangle 3. \text{ QED BY } \langle 1 \rangle 2av, \langle 7 \rangle 1, ReceivedMonotone, MessageType, \langle 3 \rangle 1 \text{ DEF } MsgInv2av, TypeOK \\
\langle 3 \rangle 20. \text{ QED BY } \langle 1 \rangle 2av, \langle 2 \rangle 4, \langle 3 \rangle 6, MessageType, ReceivedMonotone \text{ DEF } MsgInv2av, AcceptorReceiveAction \\
\langle 2 \rangle 5. \text{ CASE } AcceptorDisconnectAction \text{ BY } \langle 1 \rangle 2av, \langle 2 \rangle 5 \text{ DEF } AcceptorDisconnectAction, Disconnect, MsgInv2av \\
\langle 2 \rangle 6. \text{ CASE } LearnerAction \text{ BY } \langle 1 \rangle 2av, \langle 2 \rangle 6 \text{ DEF } LearnerAction, LearnerRecv, LearnerDecide, MsgInv2av, Next \\
\langle 2 \rangle 7. \text{ CASE } FakeAcceptorAction \\
& \quad \text{BY } \langle 1 \rangle 2av, \langle 2 \rangle 7, SafeAcceptorAssumption \text{ DEF } FakeAcceptorAction, FakeSend, MsgInv2av, Send \\
\langle 2 \rangle 8. \text{ QED BY } \langle 1 \rangle 2av, \langle 2 \rangle 0b, \langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 4, \langle 2 \rangle 5, \langle 2 \rangle 6, \langle 2 \rangle 7 \text{ DEF } Next \\
\langle 1 \rangle 2b. \text{ ASSUME } TypeOK, Next, \forall m \in msgs : m.acc \in SafeAcceptor \wedge m.type = \text{"2b"} \Rightarrow MsgInv2b(m), \\
& \quad \text{NEW } m \in msgs', m.acc \in SafeAcceptor, m.type = \text{"2b"} \\
& \quad \text{PROVE } MsgInv2b(m)' \\
\langle 2 \rangle 0a. TypeOK \text{ BY } \langle 1 \rangle 2b \\
\langle 2 \rangle 0b. TypeOK' \text{ BY } \langle 1 \rangle 2b, TypeOKInvariant \\
\langle 2 \rangle 0c. m \in Message \text{ BY } \langle 2 \rangle 0b \text{ DEF } TypeOK \\
\langle 2 \rangle 0d. m.acc \in SafeAcceptor \text{ BY } \langle 1 \rangle 2b \\
\langle 2 \rangle 0e. m.type = \text{"2b"} \text{ BY } \langle 1 \rangle 2b \\
\langle 2 \rangle 1. \text{ CASE } ProposerAction \\
& \quad \langle 3 \rangle 1. m \in msgs \text{ BY } \langle 2 \rangle 1, \langle 2 \rangle 0e \text{ DEF } ProposerAction, Phase1a, Phase1c, Send \\
& \quad \langle 3 \rangle 10. \text{ QED BY } \langle 1 \rangle 2b, \langle 2 \rangle 1, \langle 2 \rangle 0a, \langle 2 \rangle 0b, \langle 2 \rangle 0d, \langle 2 \rangle 0e, \langle 3 \rangle 1 \\
& \quad \text{DEF } TypeOK, ProposerAction, Phase1a, Phase1c, MsgInv2b, Next, Send \\
\langle 2 \rangle 2. \text{ CASE } AcceptorSendAction \\
\langle 3 \rangle \text{ HIDE DEF } Next \\
\langle 3 \rangle \text{ SUFFICES ASSUME NEW } lrn \in Learner, \\
& \quad \text{NEW } bal \in Ballot, \\
& \quad \text{NEW } acc \in SafeAcceptor, \\
& \quad \text{NEW } val \in Value, \\
& \quad \vee Phase1b(lrn, bal, acc) \\
& \quad \vee Phase2av(lrn, bal, acc, val) \\
& \quad \vee Phase2b(lrn, bal, acc, val) \\
& \quad \text{PROVE } MsgInv2b(m)'
\end{aligned}$$

BY $\langle 2 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 3 \rangle 1$. CASE *Phase1b*(*lrn*, *bal*, *acc*)
 $\langle 4 \rangle 1$. $m \in \text{msgs}$ BY $\langle 3 \rangle 1$, $\langle 2 \rangle 0a$, $\langle 2 \rangle 0e$ DEF *Phase1b*, *Send*, *TypeOK*
 $\langle 4 \rangle 2$. QED BY $\langle 1 \rangle 2b$, $\langle 3 \rangle 1$, $\langle 2 \rangle 0a$, $\langle 2 \rangle 0b$, $\langle 2 \rangle 0e$, $\langle 4 \rangle 1$ DEF *Phase1b*, *MsgInv2b*, *Send*, *TypeOK*
 $\langle 3 \rangle 2$. CASE *Phase2av*(*lrn*, *bal*, *acc*, *val*)
 $\langle 4 \rangle 1$. $m \in \text{msgs}$ BY $\langle 3 \rangle 2$, $\langle 2 \rangle 0a$, $\langle 2 \rangle 0e$ DEF *Phase2av*, *Send*, *TypeOK*
 $\langle 4 \rangle 2$. QED BY $\langle 1 \rangle 2b$, $\langle 3 \rangle 2$, $\langle 2 \rangle 0a$, $\langle 2 \rangle 0d$, $\langle 2 \rangle 0e$, $\langle 4 \rangle 1$ DEF *Phase2av*, *MsgInv2b*, *Send*, *TypeOK*
 $\langle 3 \rangle 3$. CASE *Phase2b*(*lrn*, *bal*, *acc*, *val*)
 $\langle 4 \rangle 1$. CASE $m \in \text{msgs}$
 $\langle 5 \rangle 1$. $([lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in \text{votesSent}[m.acc])'$
 BY $\langle 1 \rangle 2b$, $\langle 2 \rangle 0e$, $\langle 4 \rangle 1$, *MessageType*, *VotesSentMonotone* DEF *MsgInv2b*, *TypeOK*
 $\langle 5 \rangle 2$. $(\exists Q \in \text{ByzQuorum} :$
 $\quad \wedge [lr \mapsto m.lr, q \mapsto Q] \in \text{TrustLive}$
 $\quad \wedge \forall ba \in Q :$
 $\quad \quad \exists m2av \in \text{received}[m.acc] :$
 $\quad \quad \quad \wedge m2av.type = \text{"2av"}$
 $\quad \quad \quad \wedge m2av.lr = m.lr$
 $\quad \quad \quad \wedge m2av.acc = ba$
 $\quad \quad \quad \wedge m2av.bal = m.bal$
 $\quad \quad \quad \wedge m2av.val = m.val)'$
 BY $\langle 1 \rangle 2b$, $\langle 3 \rangle 3$, $\langle 2 \rangle 0a$, $\langle 2 \rangle 0b$, $\langle 2 \rangle 0d$, $\langle 2 \rangle 0e$, $\langle 4 \rangle 1$ DEF *Phase2b*, *MsgInv2b*, *Send*, *TypeOK*
 $\langle 5 \rangle 3$. QED BY $\langle 5 \rangle 1$, $\langle 5 \rangle 2$ DEF *MsgInv2b*
 $\langle 4 \rangle 2$. CASE $m \notin \text{msgs}$
 $\langle 5 \rangle$ SUFFICES
 ASSUME NEW $Q \in \text{ByzQuorum}$,
 $[lr \mapsto lrn, q \mapsto Q] \in \text{TrustLive}$,
 $\forall aa \in Q :$
 $\quad \exists m_1 \in \{mm \in \text{received}[acc] :$
 $\quad \quad \wedge mm.type = \text{"2av"}$
 $\quad \quad \wedge mm.lr = lrn$
 $\quad \quad \wedge mm.bal = bal\} :$
 $\quad \quad \wedge m_1.val = val$
 $\quad \quad \wedge m_1.acc = aa,$
 $\quad \text{Send}([type \mapsto \text{"2b"}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]),$
 $\quad \text{votesSent}' = [\text{votesSent} \text{ EXCEPT } ![acc] =$
 $\quad \quad \text{votesSent}[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
 PROVE *MsgInv2b*(m)'
 BY $\langle 3 \rangle 3$ DEF *Phase2b*
 $\langle 5 \rangle 1$. $m = [type \mapsto \text{"2b"}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]$ BY $\langle 4 \rangle 2$ DEF *Send*
 $\langle 5 \rangle 1e$. UNCHANGED *received* BY $\langle 3 \rangle 3$ DEF *Phase2b*
 $\langle 5 \rangle 2$. $([lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in \text{votesSent}[m.acc])'$
 BY $\langle 5 \rangle 1$, $\langle 2 \rangle 0a$, $\langle 2 \rangle 0b$, $\langle 2 \rangle 0e$, *MessageType* DEF *TypeOK*
 $\langle 5 \rangle 3$. $(\exists Q_1 \in \text{ByzQuorum} :$
 $\quad \wedge [lr \mapsto m.lr, q \mapsto Q_1] \in \text{TrustLive}$
 $\quad \wedge \forall ba \in Q_1 :$

$\exists m2av \in received[m.acc] :$
 $\wedge m2av.type = "2av"$
 $\wedge m2av.lr = m.lr$
 $\wedge m2av.acc = ba$
 $\wedge m2av.bal = m.bal$
 $\wedge m2av.val = m.val)'$
 $\langle 6 \rangle 1.$ WITNESS $Q \in ByzQuorum$
 $\langle 6 \rangle 2.$ QED BY $\langle 5 \rangle 1, \langle 5 \rangle 1e, \langle 2 \rangle 0a$ DEF *Send, TypeOK*
 $\langle 5 \rangle 4.$ QED BY $\langle 5 \rangle 2, \langle 5 \rangle 3$ DEF *MsgInv2b*
 $\langle 4 \rangle 3.$ QED BY $\langle 4 \rangle 1, \langle 4 \rangle 2$
 $\langle 3 \rangle 5.$ QED BY $\langle 3 \rangle 1, \langle 3 \rangle 2, \langle 3 \rangle 3$
 $\langle 2 \rangle 4.$ CASE *AcceptorReceiveAction*
 $\langle 3 \rangle 0.$ SUFFICES ASSUME NEW $lrn \in Learner,$
NEW $acc \in Acceptor,$
NEW $m0 \in msgs,$
 $received' = [received \text{ EXCEPT } ![acc] = received[acc] \cup \{m0\}],$
UNCHANGED $\langle msgs, maxBal, 2avSent, votesSent, connected,$
 $receivedByLearner, decision \rangle$
PROVE $MsgInv2b(m)'$
BY $\langle 2 \rangle 4, \langle 2 \rangle 0b$ DEF *AcceptorReceiveAction, Recv, TypeOK*
 $\langle 3 \rangle 2.$ $m \in msgs$ BY $\langle 3 \rangle 0, \langle 1 \rangle 2b$
 $\langle 3 \rangle 2a.$ $m \in Message$ BY $\langle 3 \rangle 2, \langle 1 \rangle 2b$ DEF *TypeOK*
 $\langle 3 \rangle 2b.$ *TypeOK* BY $\langle 1 \rangle 2b$ DEF *Phase2b*
 $\langle 3 \rangle 2c.$ *TypeOK'* BY $\langle 1 \rangle 2b, \langle 3 \rangle 2b, TypeOKInvariant$
 $\langle 3 \rangle 3.$ $[lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in votesSent'[m.acc]$ BY $\langle 3 \rangle 0, \langle 1 \rangle 2b$ DEF *MsgInv2b*
 $\langle 3 \rangle 5.$ PICK $Q0 \in ByzQuorum :$
 $\wedge [lr \mapsto m.lr, q \mapsto Q0] \in TrustLive$
 $\wedge \forall ba \in Q0 :$
 $\exists m2av \in received[m.acc] :$
 $\wedge m2av.type = "2av"$
 $\wedge m2av.lr = m.lr$
 $\wedge m2av.acc = ba$
 $\wedge m2av.bal = m.bal$
 $\wedge m2av.val = m.val$ BY $\langle 1 \rangle 2b, \langle 2 \rangle 0e, \langle 3 \rangle 2$ DEF *MsgInv2b*
 $\langle 3 \rangle 7.$ $(\exists Q \in ByzQuorum :$
 $\wedge [lr \mapsto m.lr, q \mapsto Q] \in TrustLive$
 $\wedge \forall ba \in Q :$
 $\exists m2av \in received[m.acc] :$
 $\wedge m2av.type = "2av"$
 $\wedge m2av.lr = m.lr$
 $\wedge m2av.acc = ba$
 $\wedge m2av.bal = m.bal$
 $\wedge m2av.val = m.val)'$
 $\langle 4 \rangle 0.$ WITNESS $Q0 \in ByzQuorum$
 $\langle 4 \rangle 1.$ QED BY $\langle 1 \rangle 2b, \langle 3 \rangle 5, \langle 3 \rangle 2b, \langle 3 \rangle 2c, MessageType, ReceivedMonotone$ DEF *TypeOK*

$\langle 3 \rangle 8.$ QED BY $\langle 3 \rangle 3, \langle 3 \rangle 7$ DEF *MsgInv2b*
 $\langle 2 \rangle 5.$ CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 2b, \langle 2 \rangle 5$ DEF *AcceptorDisconnectAction, Disconnect, MsgInv2b,*
 $\langle 2 \rangle 6.$ CASE *LearnerAction* BY $\langle 1 \rangle 2b, \langle 2 \rangle 6$ DEF *LearnerAction, LearnerRecv, LearnerDecide, MsgInv2b, Next*
 $\langle 2 \rangle 7.$ CASE *FakeAcceptorAction*
BY $\langle 1 \rangle 2b, \langle 2 \rangle 7, \text{SafeAcceptorAssumption}$ DEF *FakeAcceptorAction, FakeSend, MsgInv2b, Send*
 $\langle 2 \rangle 8.$ QED BY $\langle 1 \rangle 2b, \langle 2 \rangle 0a, \langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 4, \langle 2 \rangle 5, \langle 2 \rangle 6, \langle 2 \rangle 7$ DEF *Next*
 $\langle 1 \rangle 3.$ QED BY $\langle 1 \rangle 1b, \langle 1 \rangle 2av, \langle 1 \rangle 2b$

$\text{CannotDecide}(Q, L, B, V) \triangleq$
 $\exists A \in \text{SafeAcceptor} :$
 $\wedge A \in Q$
 $\wedge \exists L0 \in \text{Learner} : \text{LeftBallot}(L0, A, B)$ TODO: check if used
 $\wedge \neg \text{VotedFor}(L, A, B, V)$

$\text{HeterogeneousSpec} \triangleq$
 $\forall L1, L2 \in \text{Learner} :$
 $\forall B1, B2 \in \text{Ballot} :$
 $\forall V1, V2 \in \text{Value} :$
 $\forall A2 \in \text{SafeAcceptor} :$
 $\forall Q \in \text{ByzQuorum} :$
 $\forall M \in \text{msgs} :$
 $\wedge \langle L1, L2 \rangle \in \text{Ent}$
 $\wedge [lr \mapsto L1, q \mapsto Q] \in \text{TrustLive}$
 $\wedge M.\text{type} = \text{"2av"} \wedge M.lr = L2 \wedge M.\text{acc} = A2 \wedge M.\text{bal} = B2 \wedge M.\text{val} = V2$
 $\wedge B1 < B2$
 $\wedge V1 \neq V2$
 \Rightarrow
 $\text{CannotDecide}(Q, L1, B1, V1)$

LEMMA $\text{HeterogeneousSpecInvariant} \triangleq$
 $\text{TypeOK} \wedge \text{Next} \wedge \text{ReceivedSpec} \wedge$
 $\text{2avSentSpec1} \wedge$
 $\text{VotesSentSpec2} \wedge \text{VotesSentSpec3} \wedge \text{VotesSentSpec4} \wedge$
 $\text{ConnectedSpec} \wedge \text{MsgInv} \wedge$
 $\text{HeterogeneousSpec} \Rightarrow \text{HeterogeneousSpec}'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME $\text{TypeOK}, \text{Next}, \text{ReceivedSpec}, \text{2avSentSpec1}, \text{VotesSentSpec2}, \text{VotesSentSpec3}, \text{VotesSentSpec4},$
 $\text{ConnectedSpec}, \text{MsgInv}, \text{HeterogeneousSpec},$
NEW $L1 \in \text{Learner},$ NEW $L2 \in \text{Learner},$
NEW $B1 \in \text{Ballot},$ NEW $B2 \in \text{Ballot},$
NEW $V1 \in \text{Value},$ NEW $V2 \in \text{Value},$
NEW $A2 \in \text{SafeAcceptor},$
NEW $Q1 \in \text{ByzQuorum},$
NEW $m \in \text{msgs}',$
 $\langle L1, L2 \rangle \in \text{Ent},$
 $[lr \mapsto L1, q \mapsto Q1] \in \text{TrustLive},$

$$\begin{aligned}
& m.type = \text{"2av"}, m.lr = L2, m.acc = A2, m.bal = B2, m.val = V2, \\
& B1 < B2, \\
& V1 \neq V2 \\
& \text{PROVE } CannotDecide(Q1, L1, B1, V1)' \\
& \text{BY DEF HeterogeneousSpec} \\
\langle 1 \rangle \text{ USE DEF MsgInv} \\
\langle 1 \rangle 0a. TypeOK \text{ OBVIOUS} \\
\langle 1 \rangle 0b. TypeOK' \text{ BY TypeOKInvariant} \\
\langle 1 \rangle 0c. m \in Message \text{ BY } \langle 1 \rangle 0b \text{ DEF TypeOK} \\
\langle 1 \rangle 1. \text{CASE ProposerAction BY } \langle 1 \rangle 1 \text{ DEF ProposerAction, Phase1a, Phase1c, Next, Send, HeterogeneousSpec} \\
\langle 1 \rangle 2. \text{CASE AcceptorSendAction} \\
\langle 2 \rangle \text{ SUFFICES ASSUME NEW } lrn \in Learner, \\
\quad \text{NEW } bal \in Ballot, \\
\quad \text{NEW } acc \in SafeAcceptor, \\
\quad \text{NEW } val \in Value, \\
\quad \vee Phase1b(lrn, bal, acc) \\
\quad \vee Phase2av(lrn, bal, acc, val) \\
\quad \vee Phase2b(lrn, bal, acc, val) \\
& \text{PROVE } CannotDecide(Q1, L1, B1, V1)' \\
& \text{BY } \langle 1 \rangle 2 \text{ DEF AcceptorSendAction} \\
\langle 2 \rangle 1. \text{CASE Phase1b}(lrn, bal, acc) \\
\langle 3 \rangle 1. m \in msgs \text{ BY } \langle 2 \rangle 1, \langle 1 \rangle 0b \text{ DEF Phase1b, Send, TypeOK, Message} \\
\langle 3 \rangle 2. \text{QED BY } \langle 3 \rangle 1 \text{ DEF HeterogeneousSpec} \\
\langle 2 \rangle 2. \text{CASE Phase2av}(lrn, bal, acc, val) \\
\langle 3 \rangle 0. msgs \subseteq msgs' \text{ BY } \langle 2 \rangle 2 \text{ DEF Phase2av, Send} \\
\langle 3 \rangle 1. \text{CASE } m \in msgs \text{ BY } \langle 3 \rangle 1 \text{ DEF HeterogeneousSpec} \\
\langle 3 \rangle 2. \text{CASE } m \notin msgs \\
\langle 4 \rangle 0. m = [type \mapsto \text{"2av"}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val] \\
\quad \text{BY } \langle 3 \rangle 2, \langle 2 \rangle 2 \text{ DEF Phase2av, Send} \\
\langle 4 \rangle 0a. lrn = L2 \wedge acc = A2 \wedge bal = B2 \wedge val = V2 \text{ BY } \langle 4 \rangle 0 \\
\langle 4 \rangle 1. maxBal[L2, A2] \leq B2 \text{ BY } \langle 2 \rangle 2, \langle 4 \rangle 0a \text{ DEF Phase2av} \\
\langle 4 \rangle 2. KnowsSafeAt(L2, A2, B2, V2) \text{ BY } \langle 2 \rangle 2, \langle 4 \rangle 0a \text{ DEF Phase2av} \\
\langle 4 \rangle 3a. \text{CASE KnowsSafeAt1}(L2, A2, B2, V2) \\
\langle 5 \rangle 0. \text{USE DEF CannotDecide} \\
\langle 5 \rangle 1. \text{PICK } Q2 \in ByzQuorum : \\
\quad \wedge [lr \mapsto L2, q \mapsto Q2] \in TrustLive \\
\quad \wedge \forall a \in Q2 : \\
\quad \quad \exists m1b \in received[A2] : \\
\quad \quad \quad \wedge m1b.type = \text{"1b"} \\
\quad \quad \quad \wedge m1b.lr = L2 \\
\quad \quad \quad \wedge m1b.bal = B2 \\
\quad \quad \quad \wedge m1b.acc = a \\
\quad \quad \quad \wedge \forall p \in \{pp \in m1b.votes : \langle pp.lr, L2 \rangle \in connected[A2]\} : \\
\quad \quad \quad \quad B2 \leq p.bal \\
& \text{BY } \langle 4 \rangle 3a \text{ DEF KnowsSafeAt1}
\end{aligned}$$

$\langle 5 \rangle 2.$ PICK $S \in \text{SafeAcceptor} : S \in Q1 \wedge S \in Q2$ BY *EntanglementTrustLive*, $\langle 4 \rangle 0$, $\langle 5 \rangle 1$
 $\langle 5 \rangle 3.$ PICK $m1b \in \text{received}[A2] :$
 $\quad \wedge m1b.type = \text{"1b"}$
 $\quad \wedge m1b.lr = L2$
 $\quad \wedge m1b.bal = B2$
 $\quad \wedge m1b.acc = S$
 $\quad \wedge \forall p \in \{pp \in m1b.votes : \langle pp.lr, L2 \rangle \in \text{connected}[A2]\} :$
 $\quad \quad B2 \leq p.bal$
BY $\langle 5 \rangle 1$, $\langle 5 \rangle 2$
 $\langle 5 \rangle 4.$ $\wedge m1b \in \text{msgs}$
 $\quad \wedge m1b.type = \text{"1b"}$
 $\quad \wedge m1b.lr = L2$
 $\quad \wedge m1b.bal = B2$
 $\quad \wedge m1b.acc = S$
 $\quad \wedge \forall p \in \{pp \in m1b.votes : \langle pp.lr, L2 \rangle \in \text{connected}[A2]\} :$
 $\quad \quad B2 \leq p.bal$
BY $\langle 5 \rangle 3$, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *ReceivedSpec*
 $\langle 5 \rangle 5.$ WITNESS $S \in \text{SafeAcceptor}$
 $\langle 5 \rangle 6.$ $\exists L \in \text{Learner} : \text{LeftBallot}(L, S, B1)$ BY $\langle 5 \rangle 4$, $\langle 3 \rangle 0$ DEF *LeftBallot*
 $\langle 5 \rangle 7.$ $\neg \text{VotedFor}(L1, S, B1, V1)'$
 $\langle 6 \rangle 1.$ SUFFICES ASSUME *VotedFor*($L1, S, B1, V1$) PROVE FALSE OBVIOUS
 $\langle 6 \rangle 2.$ $[lr \mapsto L1, bal \mapsto B1, val \mapsto V1] \in \text{votesSent}[S]$ BY $\langle 6 \rangle 1$ DEF *VotesSentSpec2*
 $\langle 6 \rangle 3.$ $m1b.votes = \{p \in \text{votesSent}[S] : \text{MaxVote}(S, B2, p)\}$ BY $\langle 5 \rangle 4$ DEF *MsgInv1b*
 $\langle 6 \rangle 4.$ PICK $P \in \text{votesSent}[S] : \text{MaxVote}(S, B2, P) \wedge P.lr = L1 \wedge B1 \leq P.bal$
 $\langle 7 \rangle 1.$ SUFFICES ASSUME NEW $P0 \in \text{votesSent}[S]$,
 $\quad P0 = [lr \mapsto L1, bal \mapsto B1, val \mapsto V1]$
PROVE $\exists P \in \text{votesSent}[S] : \text{MaxVote}(S, B2, P) \wedge P.lr = P0.lr \wedge P0.bal \leq P.bal$
BY $\langle 6 \rangle 2$
 $\langle 7 \rangle 2.$ $P0.bal < B2$ BY $\langle 7 \rangle 1$
 $\langle 7 \rangle 3.$ QED BY $\langle 7 \rangle 1$, $\langle 7 \rangle 2$ DEF *VotesSentSpec3*
 $\langle 6 \rangle 5.$ $P \in m1b.votes$ BY $\langle 6 \rangle 3$, $\langle 6 \rangle 4$
 $\langle 6 \rangle 6.$ $\langle P.lr, L2 \rangle \in \text{connected}[A2]$ BY $\langle 6 \rangle 4$ DEF *ConnectedSpec*
 $\langle 6 \rangle 7.$ $B2 \leq P.bal$ BY $\langle 6 \rangle 5$, $\langle 6 \rangle 6$, $\langle 5 \rangle 4$
 $\langle 6 \rangle 8.$ $P \in [lr : \text{Learner}, bal : \text{Ballot}, val : \text{Value}]$ BY $\langle 6 \rangle 4$, *SafeAcceptorIsAcceptor* DEF *TypeOK*
 $\langle 6 \rangle 9.$ $P.bal \in \text{Ballot}$ BY $\langle 6 \rangle 8$
 $\langle 6 \rangle 10.$ QED BY $\langle 6 \rangle 9$, $\langle 6 \rangle 7$, $\langle 6 \rangle 4$, *BallotLeNotLeq* DEF *MaxVote*
 $\langle 5 \rangle 8.$ QED BY $\langle 5 \rangle 2$, $\langle 5 \rangle 6$, $\langle 5 \rangle 7$
 $\langle 4 \rangle 3b.$ CASE *KnowsSafeAt2*($L2, A2, B2, V2$)
 $\langle 5 \rangle 1.$ PICK $c \in \text{Ballot}$, $BQ \in \text{ByzQuorum}$, $WQ \in \text{ByzQuorum} :$
 $\quad \wedge c < B2$
 $\quad \wedge [lr \mapsto L2, q \mapsto BQ] \in \text{TrustLive}$
 $\quad \wedge \forall a \in BQ :$
 $\quad \quad \exists m1 \in \{mm \in \text{received}[A2] : mm.type = \text{"1b"} \wedge mm.lr = L2 \wedge mm.bal = B2\} :$
 $\quad \quad \wedge m1.acc = a$
 $\quad \quad \wedge \forall p \in \{pp \in m1.votes : \langle pp.lr, L2 \rangle \in \text{connected}[A2]\} :$

$$\begin{aligned}
& \wedge p.bal \leq c \\
& \wedge (p.bal = c) \Rightarrow (p.val = V2) \\
& \wedge [lr \mapsto L2, q \mapsto WQ] \in TrustLive \\
& \wedge \forall a \in WQ : \\
& \quad \exists m2 \in \{mm \in received[A2] : mm.type = \text{"1b"} \wedge mm.lr = L2 \wedge mm.bal = B2\} : \\
& \quad \wedge m2.acc = a \\
& \quad \wedge \exists p \in m2.proposals : \\
& \quad \quad \wedge p.lr = L2 \\
& \quad \quad \wedge p.bal = c \\
& \quad \quad \wedge p.val = V2 \\
& \text{BY } \langle 4 \rangle 3b, \langle 4 \rangle 0a \text{ DEF } KnowsSafeAt2, Ballot \\
& \langle 5 \rangle 2. \text{ PICK } S1 \in SafeAcceptor : S1 \in Q1 \wedge S1 \in BQ \text{ BY } EntanglementTrustLive, \langle 4 \rangle 0, \langle 5 \rangle 1 \\
& \langle 5 \rangle 4. \text{ PICK } m1 \in received[A2] : \\
& \quad \wedge m1.type = \text{"1b"} \\
& \quad \wedge m1.lr = L2 \\
& \quad \wedge m1.bal = B2 \\
& \quad \wedge m1.acc = S1 \\
& \quad \wedge \forall p \in \{pp \in m1.votes : \langle pp.lr, L2 \rangle \in connected[A2]\} : \\
& \quad \quad \wedge p.bal \leq c \\
& \quad \quad \wedge p.bal = c \Rightarrow p.val = V2 \\
& \text{BY } \langle 5 \rangle 1, \langle 5 \rangle 2 \\
& \langle 5 \rangle 5. \wedge m1 \in msgs \\
& \quad \wedge m1.type = \text{"1b"} \\
& \quad \wedge m1.lr = L2 \\
& \quad \wedge m1.bal = B2 \\
& \quad \wedge m1.acc = S1 \\
& \quad \wedge \forall p \in \{pp \in m1.votes : \langle pp.lr, L2 \rangle \in connected[A2]\} : \\
& \quad \quad \wedge p.bal \leq c \\
& \quad \quad \wedge p.bal = c \Rightarrow p.val = V2 \\
& \text{BY } \langle 5 \rangle 4, SafeAcceptorIsAcceptor \text{ DEF } TypeOK, ReceivedSpec \\
& \langle 5 \rangle 6. \text{ CASE } \neg VotedFor(L1, S1, B1, V1) \\
& \quad \langle 6 \rangle 1. \neg VotedFor(L1, S1, B1, V1)' \text{ BY } \langle 5 \rangle 6, \langle 2 \rangle 2 \text{ DEF } VotedFor, Phase2av, Send \\
& \quad \langle 6 \rangle 2. \text{ QED BY } \langle 6 \rangle 1, \langle 5 \rangle 2, \langle 5 \rangle 5, MsgsMonotone \text{ DEF } LeftBallot, CannotDecide \\
& \langle 5 \rangle 7. \text{ CASE } VotedFor(L1, S1, B1, V1) \\
& \quad \langle 6 \rangle 1. [lr \mapsto L1, bal \mapsto B1, val \mapsto V1] \in votesSent[S1] \text{ BY } \langle 5 \rangle 7 \text{ DEF } VotesSentSpec2 \\
& \quad \langle 6 \rangle 2. \text{ PICK } P \in votesSent[S1] : MaxVote(S1, B2, P) \wedge P.lr = L1 \wedge B1 \leq P.bal \\
& \quad \quad \langle 7 \rangle 1. \text{ SUFFICES ASSUME NEW } vote \in votesSent[S1], vote = [lr \mapsto L1, bal \mapsto B1, val \mapsto V1] \\
& \quad \quad \quad \text{PROVE } \exists P \in votesSent[S1] : MaxVote(S1, B2, P) \wedge P.lr = L1 \wedge vote.bal \leq P.bal \\
& \quad \quad \quad \text{BY } \langle 6 \rangle 1 \\
& \quad \quad \langle 7 \rangle 2. \text{ QED BY } \langle 7 \rangle 1, SafeAcceptorIsAcceptor \text{ DEF } VotesSentSpec3, TypeOK \\
& \langle 6 \rangle 3. P \in m1.votes \text{ BY } \langle 6 \rangle 2, \langle 5 \rangle 5 \text{ DEF } MsgInv1b \\
& \langle 6 \rangle 4. \langle P.lr, L2 \rangle \in connected[A2] \text{ BY } \langle 5 \rangle 5, \langle 6 \rangle 2 \text{ DEF } ConnectedSpec \\
& \langle 6 \rangle 5. P.bal \in Ballot \text{ BY } \langle 5 \rangle 5, \langle 6 \rangle 3, SafeAcceptorIsAcceptor, MessageType \text{ DEF } TypeOK \\
& \langle 6 \rangle 6. B1 < c \\
& \quad \langle 7 \rangle 1. \text{ CASE } P.val = V1
\end{aligned}$$

$\langle 8 \rangle 1. P.bal \leq c \wedge (P.bal = c \Rightarrow P.val = V2) \text{BY } \langle 5 \rangle 5, \langle 6 \rangle 3, \langle 6 \rangle 4$
 $\langle 8 \rangle 2. P.bal < c \text{BY } \langle 6 \rangle 5, \langle 8 \rangle 1, \langle 7 \rangle 1 \text{ DEF } Ballot$
 $\langle 8 \rangle 10. \text{QED BY } \langle 6 \rangle 2, \langle 6 \rangle 5, \langle 8 \rangle 2, BallotLeqLeTrans$
 $\langle 7 \rangle 2. \text{CASE } P.val \neq V1$
 $\langle 8 \rangle 1. B1 < P.bal$
 $\langle 9 \rangle 0. \langle L1, L1 \rangle \in Ent \text{BY } EntanglementSelf$
 $\langle 9 \rangle 1. B1 \leq P.bal \text{BY } \langle 6 \rangle 2$
 $\langle 9 \rangle 2. B1 \neq P.bal \text{BY } \langle 6 \rangle 1, \langle 6 \rangle 2, \langle 6 \rangle 5, \langle 7 \rangle 2, \langle 9 \rangle 0 \text{ DEF } VotesSentSpec4$
 $\langle 9 \rangle 3. \text{QED BY } \langle 6 \rangle 5, \langle 9 \rangle 1, \langle 9 \rangle 2 \text{ DEF } Ballot$
 $\langle 8 \rangle 2. P.bal \leq c \text{BY } \langle 5 \rangle 5, \langle 6 \rangle 3, \langle 6 \rangle 4$
 $\langle 8 \rangle 3. \text{QED BY } \langle 8 \rangle 1, \langle 8 \rangle 2, \langle 6 \rangle 5, BallotLeqLeTrans$
 $\langle 7 \rangle 3. \text{QED BY } \langle 7 \rangle 1, \langle 7 \rangle 2$
 $\langle 6 \rangle 7. \text{PICK } S2 \in SafeAcceptor : S2 \in Q1 \wedge S2 \in WQ \text{BY } EntanglementTrustLive, \langle 4 \rangle 0, \langle 5 \rangle 1$
 $\langle 6 \rangle 8. \text{PICK } m2 \in received[A2] :$
 $\quad \wedge m2.type = "1b"$
 $\quad \wedge m2.lr = L2$
 $\quad \wedge m2.bal = B2$
 $\quad \wedge m2.acc = S2$
 $\quad \wedge \exists p \in m2.proposals : p.lr = L2 \wedge p.bal = c \wedge p.val = V2$
 $\text{BY } \langle 5 \rangle 1, \langle 6 \rangle 7$
 $\langle 6 \rangle 9. \text{PICK } p2 \in m2.proposals :$
 $\quad \wedge m2 \in msgs$
 $\quad \wedge m2.type = "1b"$
 $\quad \wedge m2.lr = L2$
 $\quad \wedge m2.bal = B2$
 $\quad \wedge m2.acc = S2$
 $\quad \wedge p2.lr = L2$
 $\quad \wedge p2.bal = c$
 $\quad \wedge p2.val = V2$
 $\text{BY } \langle 6 \rangle 8, SafeAcceptorIsAcceptor \text{ DEF } TypeOK, ReceivedSpec$
 $\langle 6 \rangle 10. Proposed(L2, S2, c, V2)$
 $\langle 7 \rangle 1. p2 \in 2avSent[S2] \text{BY } \langle 6 \rangle 9 \text{ DEF } MsgInv1b$
 $\langle 7 \rangle 2. \text{QED BY } \langle 7 \rangle 1, \langle 6 \rangle 9 \text{ DEF } 2avSentSpec1$
 $\langle 6 \rangle 11. \text{PICK } m2av \in msgs :$
 $\quad \wedge m2av.type = "2av"$
 $\quad \wedge m2av.lr = L2$
 $\quad \wedge m2av.acc = S2$
 $\quad \wedge m2av.bal = c$
 $\quad \wedge m2av.val = V2$
 $\text{BY } \langle 6 \rangle 10 \text{ DEF } Proposed$
 $\langle 6 \rangle 12. \text{SUFFICES } CannotDecide(Q1, L1, B1, V1) \text{BY } \text{ DEF } CannotDecide$
 $\langle 6 \rangle 15. \text{QED BY } \langle 6 \rangle 11, \langle 6 \rangle 6 \text{ DEF } HeterogeneousSpec$
 $\langle 5 \rangle 8. \text{QED BY } \langle 5 \rangle 6, \langle 5 \rangle 7$
 $\langle 4 \rangle 4. \text{QED BY } \langle 4 \rangle 3a, \langle 4 \rangle 3b, \langle 4 \rangle 2 \text{ DEF } KnowsSafeAt$
 $\langle 3 \rangle 3. \text{QED BY } \langle 3 \rangle 1, \langle 3 \rangle 2$

$\langle 2 \rangle 3.$ CASE *Phase2b*(*lrn*, *bal*, *acc*, *val*)
 $\langle 3 \rangle 1.$ $m \in \text{msgsBY } \langle 2 \rangle 3, \langle 1 \rangle 0b$ DEF *Phase2b*, *Send*, *TypeOK*
 $\langle 3 \rangle 2.$ QED BY $\langle 3 \rangle 1$ DEF *HeterogeneousSpec*
 $\langle 2 \rangle 4.$ QED BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 3.$ CASE *AcceptorReceiveAction* BY $\langle 1 \rangle 3$ DEF *AcceptorReceiveAction*, *Next*, *Recv*, *HeterogeneousSpec*
 $\langle 1 \rangle 4.$ CASE *AcceptorDisconnectAction* BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction*, *Disconnect*, *Next*, *HeterogeneousSpec*
 $\langle 1 \rangle 5.$ CASE *LearnerAction*
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in \text{Learner}$, NEW $bal \in \text{Ballot}$,
 $\quad \vee \text{LearnerDecide}(lrn, bal)$
 $\quad \vee \text{LearnerRecv}(lrn)$
PROVE *CannotDecide*($Q1, L1, B1, V1$)'
BY $\langle 1 \rangle 5$ DEF *LearnerAction*
 $\langle 2 \rangle 2.$ CASE *LearnerDecide*(lrn, bal) BY $\langle 2 \rangle 2$ DEF *LearnerDecide*, *Next*, *HeterogeneousSpec*
 $\langle 2 \rangle 3.$ CASE *LearnerRecv*(lrn) BY $\langle 2 \rangle 2$ DEF *LearnerRecv*, *Next*, *HeterogeneousSpec*
 $\langle 2 \rangle 4.$ QED BY $\langle 2 \rangle 2, \langle 2 \rangle 3$
 $\langle 1 \rangle 6.$ CASE *FakeAcceptorAction* BY $\langle 1 \rangle 6$, *SafeAcceptorAssumption* DEF *FakeAcceptorAction*, *FakeSend*, *Send*, *HeterogeneousSpec*
 $\langle 1 \rangle 7.$ QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

LEMMA *ChosenSafeCaseEq* \triangleq
ASSUME NEW $L1 \in \text{Learner}$, NEW $L2 \in \text{Learner}$,
NEW $B \in \text{Ballot}$,
NEW $V1 \in \text{Value}$, NEW $V2 \in \text{Value}$,
TypeOK, *MsgInv*,
ReceivedSpec, *ReceivedByLearnerSpec*, *VotesSentSpec4*,
 $\langle L1, L2 \rangle \in \text{Ent}$,
ChosenIn($L1, B, V1$), *ChosenIn*($L2, B, V2$)
PROVE $V1 = V2$

PROOF
 $\langle 1 \rangle$ USE DEF *MsgInv*
 $\langle 1 \rangle 1.$ PICK $Q1 \in \text{ByzQuorum}$:
 $\quad \wedge [lr \mapsto L1, q \mapsto Q1] \in \text{TrustLive}$
 $\quad \wedge \forall aa \in Q1$:
 $\quad \quad \exists m \in \{mm \in \text{receivedByLearner}[L1] : mm.bal = B\}$:
 $\quad \quad \quad \wedge m.val = V1$
 $\quad \quad \quad \wedge m.acc = aa$
BY DEF *ChosenIn*
 $\langle 1 \rangle 2.$ PICK $Q2 \in \text{ByzQuorum}$:
 $\quad \wedge [lr \mapsto L2, q \mapsto Q2] \in \text{TrustLive}$
 $\quad \wedge \forall aa \in Q2$:
 $\quad \quad \exists m \in \{mm \in \text{receivedByLearner}[L2] : mm.bal = B\}$:
 $\quad \quad \quad \wedge m.val = V2$
 $\quad \quad \quad \wedge m.acc = aa$
BY DEF *ChosenIn*
 $\langle 1 \rangle 3.$ PICK $A \in \text{SafeAcceptor} : A \in Q1 \wedge A \in Q2$ BY *EntanglementTrustLive*, $\langle 1 \rangle 1, \langle 1 \rangle 2$
 $\langle 1 \rangle 4.$ PICK $m1 \in \text{receivedByLearner}[L1] : m1.acc = A \wedge m1.bal = B \wedge m1.val = V1$ BY $\langle 1 \rangle 1, \langle 1 \rangle 3$ DEF *ChosenIn*

$\langle 1 \rangle 5.$ PICK $m2 \in receivedByLearner[L2] : m2.acc = A \wedge m2.bal = B \wedge m2.val = V2$ BY $\langle 1 \rangle 2, \langle 1 \rangle 3$ DEF *Chosen*
 $\langle 1 \rangle 6.$ $\wedge m1 \in msgs$
 $\quad \wedge m1.type = \text{"2b"}$
 $\quad \wedge m1.lr = L1$
 $\quad \wedge m1.acc = A$
 $\quad \wedge m1.bal = B$
 $\quad \wedge m1.val = V1$
BY $\langle 1 \rangle 4$ DEF *ReceivedByLearnerSpec, TypeOK*
 $\langle 1 \rangle 7.$ $\wedge m2 \in msgs$
 $\quad \wedge m2.type = \text{"2b"}$
 $\quad \wedge m2.lr = L2$
 $\quad \wedge m2.acc = A$
 $\quad \wedge m2.bal = B$
 $\quad \wedge m2.val = V2$
BY $\langle 1 \rangle 5$ DEF *ReceivedByLearnerSpec, TypeOK*
 $\langle 1 \rangle 8.$ $[lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in votesSent[A]$ BY $\langle 1 \rangle 6$ DEF *MsgInv2b*
 $\langle 1 \rangle 9.$ $[lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in votesSent[A]$ BY $\langle 1 \rangle 7$ DEF *MsgInv2b*
 $\langle 1 \rangle 100.$ QED BY $\langle 1 \rangle 8, \langle 1 \rangle 9$ DEF *VotesSentSpec4*

LEMMA *ChosenSafeCaseLt* \triangleq

ASSUME NEW $L1 \in Learner$, NEW $L2 \in Learner$,
 NEW $B1 \in Ballot$, NEW $B2 \in Ballot$,
 NEW $V1 \in Value$, NEW $V2 \in Value$,
TypeOK, ReceivedSpec, ReceivedByLearnerSpec, MsgInv,
HeterogeneousSpec,
 $\langle L1, L2 \rangle \in Ent$,
 $B1 < B2$,
 $ChosenIn(L1, B1, V1), ChosenIn(L2, B2, V2)$

PROVE $V1 = V2$

PROOF

$\langle 1 \rangle$ USE DEF *MsgInv*
 $\langle 1 \rangle$ SUFFICES ASSUME $V1 \neq V2$ PROVE FALSE OBVIOUS
 $\langle 1 \rangle 1.$ PICK $Q1 \in ByzQuorum$:
 $\quad \wedge [lr \mapsto L1, q \mapsto Q1] \in TrustLive$
 $\quad \wedge \forall aa \in Q1$:
 $\quad \quad \exists m \in \{mm \in receivedByLearner[L1] : mm.bal = B1\}$:
 $\quad \quad \quad \wedge m.val = V1$
 $\quad \quad \quad \wedge m.acc = aa$
BY DEF *ChosenIn*
 $\langle 1 \rangle 2.$ PICK $Q2 \in ByzQuorum$:
 $\quad \wedge [lr \mapsto L2, q \mapsto Q2] \in TrustLive$
 $\quad \wedge \forall aa \in Q2$:
 $\quad \quad \exists m \in \{mm \in receivedByLearner[L2] : mm.bal = B2\}$:
 $\quad \quad \quad \wedge m.val = V2$
 $\quad \quad \quad \wedge m.acc = aa$

BY DEF *ChosenIn*
 ⟨1⟩3. PICK $A \in \text{SafeAcceptor} : A \in Q1 \wedge A \in Q2$ BY *EntanglementTrustLive*, ⟨1⟩1, ⟨1⟩2
 ⟨1⟩4. PICK $m1 \in \text{receivedByLearner}[L1] : m1.\text{acc} = A \wedge m1.\text{bal} = B1 \wedge m1.\text{val} = V1$ BY ⟨1⟩1, ⟨1⟩3 DEF *Chosen*
 ⟨1⟩5. PICK $m2 \in \text{receivedByLearner}[L2] : m2.\text{acc} = A \wedge m2.\text{bal} = B2 \wedge m2.\text{val} = V2$ BY ⟨1⟩2, ⟨1⟩3 DEF *Chosen*
 ⟨1⟩6. $\wedge m1 \in \text{msgs}$
 $\wedge m1.\text{type} = \text{"2b"}$
 $\wedge m1.\text{lr} = L1$
 $\wedge m1.\text{acc} = A$
 $\wedge m1.\text{bal} = B1$
 $\wedge m1.\text{val} = V1$
 BY ⟨1⟩4 DEF *ReceivedByLearnerSpec*, *TypeOK*
 ⟨1⟩7. $\wedge m2 \in \text{msgs}$
 $\wedge m2.\text{type} = \text{"2b"}$
 $\wedge m2.\text{lr} = L2$
 $\wedge m2.\text{acc} = A$
 $\wedge m2.\text{bal} = B2$
 $\wedge m2.\text{val} = V2$
 BY ⟨1⟩5 DEF *ReceivedByLearnerSpec*, *TypeOK*
 ⟨1⟩10. PICK $R1 \in \text{ByzQuorum} :$
 $\wedge [lr \mapsto L1, q \mapsto R1] \in \text{TrustLive}$
 BY ⟨1⟩6 DEF *MsgInv2b*
 ⟨1⟩11. PICK $R2 \in \text{ByzQuorum} :$
 $\wedge [lr \mapsto L2, q \mapsto R2] \in \text{TrustLive}$
 $\wedge \forall aa \in R2 :$
 $\exists m2av \in \text{received}[A] :$
 $\wedge m2av.\text{type} = \text{"2av"}$
 $\wedge m2av.\text{lr} = L2$
 $\wedge m2av.\text{acc} = aa$
 $\wedge m2av.\text{bal} = B2$
 $\wedge m2av.\text{val} = V2$
 BY ⟨1⟩7 DEF *MsgInv2b*
 ⟨1⟩12. PICK $A0 \in \text{SafeAcceptor} : A0 \in R1 \wedge A0 \in R2$ BY *EntanglementTrustLive*, ⟨1⟩10, ⟨1⟩11
 ⟨1⟩14. PICK $m2av2 \in \text{received}[A] :$
 $m2av2.\text{type} = \text{"2av"} \wedge m2av2.\text{lr} = L2 \wedge m2av2.\text{acc} = A0 \wedge m2av2.\text{bal} = B2 \wedge m2av2.\text{val} = V2$
 BY ⟨1⟩12, ⟨1⟩11
 ⟨1⟩16. $\wedge m2av2 \in \text{msgs}$
 $\wedge m2av2.\text{type} = \text{"2av"}$
 $\wedge m2av2.\text{lr} = L2$
 $\wedge m2av2.\text{acc} = A0$
 $\wedge m2av2.\text{bal} = B2$
 $\wedge m2av2.\text{val} = V2$
 BY ⟨1⟩14, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec*, *TypeOK*
 ⟨1⟩17. *CannotDecide*($Q1, L1, B1, V1$)
 ⟨2⟩1. $[lr \mapsto L1, q \mapsto Q1] \in \text{TrustLive}$ BY ⟨1⟩1
 ⟨2⟩5. QED BY ⟨1⟩16, ⟨2⟩1 DEF *HeterogeneousSpec*

$\langle 1 \rangle 18.$ PICK $S \in \text{SafeAcceptor} : S \in Q1 \wedge \neg \text{VotedFor}(L1, S, B1, V1)$ BY $\langle 1 \rangle 17$ DEF *CannotDecide*
 $\langle 1 \rangle 19.$ PICK $m \in \text{receivedByLearner}[L1] : m.\text{acc} = S \wedge m.\text{bal} = B1 \wedge m.\text{val} = V1$
 BY $\langle 1 \rangle 18, \langle 1 \rangle 1$ DEF *CannotDecide*
 $\langle 1 \rangle 20.$ $\wedge m \in \{mm \in \text{msgs} : mm.\text{type} = \text{"2b"}\}$
 $\wedge m.\text{lr} = L1$
 $\wedge m.\text{acc} = S$
 $\wedge m.\text{bal} = B1$
 $\wedge m.\text{val} = V1$
 BY $\langle 1 \rangle 19$ DEF *ReceivedByLearnerSpec, TypeOK*
 $\langle 1 \rangle 50.$ QED BY $\langle 1 \rangle 20, \langle 1 \rangle 18$ DEF *CannotDecide, VotedFor, ReceivedByLearnerSpec, TypeOK*

LEMMA *ChosenSafe* \triangleq

ASSUME NEW $L1 \in \text{Learner}$, NEW $L2 \in \text{Learner}$,
 NEW $B1 \in \text{Ballot}$, NEW $B2 \in \text{Ballot}$,
 NEW $V1 \in \text{Value}$, NEW $V2 \in \text{Value}$,
 TypeOK , ReceivedSpec , $\text{ReceivedByLearnerSpec}$, VotesSentSpec4 , MsgInv ,
 HeterogeneousSpec ,
 $\langle L1, L2 \rangle \in \text{Ent}$,
 $\text{ChosenIn}(L1, B1, V1)$, $\text{ChosenIn}(L2, B2, V2)$
 PROVE $V1 = V2$

PROOF

$\langle 1 \rangle$ USE DEF *MsgInv*

$\langle 1 \rangle 1.$ PICK $Q1 \in \text{ByzQuorum} :$

$\wedge [lr \mapsto L1, q \mapsto Q1] \in \text{TrustLive}$

$\wedge \forall aa \in Q1 :$

$\exists m \in \{mm \in \text{receivedByLearner}[L1] : mm.\text{bal} = B1\} :$

$\wedge m.\text{val} = V1$

$\wedge m.\text{acc} = aa$

BY DEF *ChosenIn*

$\langle 1 \rangle 2.$ PICK $Q2 \in \text{ByzQuorum} :$

$\wedge [lr \mapsto L2, q \mapsto Q2] \in \text{TrustLive}$

$\wedge \forall aa \in Q2 :$

$\exists m \in \{mm \in \text{receivedByLearner}[L2] : mm.\text{bal} = B2\} :$

$\wedge m.\text{val} = V2$

$\wedge m.\text{acc} = aa$

BY DEF *ChosenIn*

$\langle 1 \rangle 3.$ PICK $A \in \text{SafeAcceptor} : A \in Q1 \wedge A \in Q2$ BY *EntanglementTrustLive*, $\langle 1 \rangle 1$, $\langle 1 \rangle 2$

$\langle 1 \rangle 4.$ PICK $m1 \in \text{receivedByLearner}[L1] : m1.\text{acc} = A \wedge m1.\text{bal} = B1 \wedge m1.\text{val} = V1$ BY $\langle 1 \rangle 1$, $\langle 1 \rangle 3$ DEF *Chosen*

$\langle 1 \rangle 5.$ PICK $m2 \in \text{receivedByLearner}[L2] : m2.\text{acc} = A \wedge m2.\text{bal} = B2 \wedge m2.\text{val} = V2$ BY $\langle 1 \rangle 2$, $\langle 1 \rangle 3$ DEF *Chosen*

$\langle 1 \rangle 6.$ $\wedge m1 \in \text{msgs}$

$\wedge m1.\text{type} = \text{"2b"}$

$\wedge m1.\text{lr} = L1$

$\wedge m1.\text{acc} = A$

$\wedge m1.\text{bal} = B1$

$\wedge m1.\text{val} = V1$

BY $\langle 1 \rangle 4$ DEF *ReceivedByLearnerSpec*, *TypeOK*
 $\langle 1 \rangle 7. \wedge m2 \in msgs$
 $\wedge m2.type = \text{"2b"}$
 $\wedge m2.lr = L2$
 $\wedge m2.acc = A$
 $\wedge m2.bal = B2$
 $\wedge m2.val = V2$
 BY $\langle 1 \rangle 5$ DEF *ReceivedByLearnerSpec*, *TypeOK*
 $\langle 1 \rangle 8. [lr \mapsto L1, bal \mapsto B1, val \mapsto V1] \in votesSent[A]$ BY $\langle 1 \rangle 6$ DEF *MsgInv2b*
 $\langle 1 \rangle 9. [lr \mapsto L2, bal \mapsto B2, val \mapsto V2] \in votesSent[A]$ BY $\langle 1 \rangle 7$ DEF *MsgInv2b*
 $\langle 1 \rangle 10. \text{PICK } R1 \in ByzQuorum :$
 $\wedge [lr \mapsto L1, q \mapsto R1] \in TrustLive$
 $\wedge \forall aa \in R1 :$
 $\quad \exists m2av \in received[A] :$
 $\quad \wedge m2av.type = \text{"2av"}$
 $\quad \wedge m2av.lr = L1$
 $\quad \wedge m2av.acc = aa$
 $\quad \wedge m2av.bal = B1$
 $\quad \wedge m2av.val = V1$
 BY $\langle 1 \rangle 6$ DEF *MsgInv2b*
 $\langle 1 \rangle 11. \text{PICK } R2 \in ByzQuorum :$
 $\wedge [lr \mapsto L2, q \mapsto R2] \in TrustLive$
 $\wedge \forall aa \in R2 :$
 $\quad \exists m2av \in received[A] :$
 $\quad \wedge m2av.type = \text{"2av"}$
 $\quad \wedge m2av.lr = L2$
 $\quad \wedge m2av.acc = aa$
 $\quad \wedge m2av.bal = B2$
 $\quad \wedge m2av.val = V2$
 BY $\langle 1 \rangle 7$ DEF *MsgInv2b*
 $\langle 1 \rangle 12. \text{PICK } A0 \in SafeAcceptor : A0 \in R1 \wedge A0 \in R2$ BY *EntanglementTrustLive*, $\langle 1 \rangle 10$, $\langle 1 \rangle 11$
 $\langle 1 \rangle 13. \text{PICK } m2av1 \in received[A] :$
 $\quad m2av1.type = \text{"2av"} \wedge m2av1.lr = L1 \wedge m2av1.acc = A0 \wedge m2av1.bal = B1 \wedge m2av1.val = V1$
 BY $\langle 1 \rangle 12$, $\langle 1 \rangle 10$
 $\langle 1 \rangle 14. \text{PICK } m2av2 \in received[A] :$
 $\quad m2av2.type = \text{"2av"} \wedge m2av2.lr = L2 \wedge m2av2.acc = A0 \wedge m2av2.bal = B2 \wedge m2av2.val = V2$
 BY $\langle 1 \rangle 12$, $\langle 1 \rangle 11$
 $\langle 1 \rangle 15. \wedge m2av1 \in msgs$
 $\wedge m2av1.type = \text{"2av"}$
 $\wedge m2av1.lr = L1$
 $\wedge m2av1.acc = A0$
 $\wedge m2av1.bal = B1$
 $\wedge m2av1.val = V1$
 BY $\langle 1 \rangle 13$, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec*, *TypeOK*
 $\langle 1 \rangle 16. \wedge m2av2 \in msgs$

$\wedge m2av2.type = \text{"2av"}$
 $\wedge m2av2.lr = L2$
 $\wedge m2av2.acc = A0$
 $\wedge m2av2.bal = B2$
 $\wedge m2av2.val = V2$
 BY $\langle 1 \rangle 14$, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec*, *TypeOK*
 $\langle 1 \rangle 30$.CASE $B1 < B2$ BY $\langle 1 \rangle 30$, *ChosenSafeCaseLt*
 $\langle 1 \rangle 31$.CASE $B2 < B1$ BY $\langle 1 \rangle 31$, *ChosenSafeCaseLt*, *EntanglementSym*
 $\langle 1 \rangle 32$.CASE $B1 = B2$ BY $\langle 1 \rangle 32$, *ChosenSafeCaseEq*
 $\langle 1 \rangle 33$. QED BY $\langle 1 \rangle 30$, $\langle 1 \rangle 31$, $\langle 1 \rangle 32$, *BallotOrderCases*

Safety \triangleq
 $\forall L1, L2 \in \text{Learner} : \forall B1, B2 \in \text{Ballot} : \forall V1, V2 \in \text{Value} :$
 $\langle L1, L2 \rangle \in \text{Ent} \wedge$
 $V1 \in \text{decision}[L1, B1] \wedge V2 \in \text{decision}[L2, B2] \Rightarrow V1 = V2$

LEMMA *SafetyStep* \triangleq
 $\text{TypeOK} \wedge \text{Next} \wedge \text{MsgInv} \wedge$
 $\text{DecisionSpec} \wedge \text{ReceivedSpec} \wedge \text{ReceivedByLearnerSpec} \wedge$
 $2avSentSpec1 \wedge 2avSentSpec3 \wedge \text{VotesSentSpec4} \wedge$
 $\text{HeterogeneousSpec} \wedge \text{Safety} \Rightarrow \text{Safety}'$

PROOF

$\langle 1 \rangle$ SUFFICES

ASSUME *TypeOK*, *Next*, *MsgInv*, *Safety*, *DecisionSpec*, *ReceivedSpec*, *ReceivedByLearnerSpec*,
2avSentSpec1, *2avSentSpec3*, *VotesSentSpec4*,
HeterogeneousSpec,
 NEW $L1 \in \text{Learner}$, NEW $L2 \in \text{Learner}$,
 NEW $B1 \in \text{Ballot}$, NEW $B2 \in \text{Ballot}$,
 NEW $V1 \in \text{Value}$, NEW $V2 \in \text{Value}$,
 $\langle L1, L2 \rangle \in \text{Ent}$,
 $V1 \in \text{decision}'[L1, B1]$, $V2 \in \text{decision}'[L2, B2]$

PROVE $V1 = V2$

BY DEF *Safety*

$\langle 1 \rangle 0a$. *TypeOK* OBVIOUS

$\langle 1 \rangle 0b$. *TypeOK'* BY *TypeOKInvariant*

$\langle 1 \rangle 1$.CASE *ProposerAction* BY $\langle 1 \rangle 1$ DEF *ProposerAction*, *Phase1a*, *Phase1c*, *Send*, *Safety*

$\langle 1 \rangle 2$.CASE *AcceptorSendAction*

$\langle 2 \rangle$ SUFFICES ASSUME NEW $lrn \in \text{Learner}$,
 NEW $bal \in \text{Ballot}$,
 NEW $acc \in \text{SafeAcceptor}$,
 NEW $val \in \text{Value}$,
 $\vee \text{Phase1b}(lrn, bal, acc)$
 $\vee \text{Phase2av}(lrn, bal, acc, val)$
 $\vee \text{Phase2b}(lrn, bal, acc, val)$

PROVE $V1 = V2$

BY $\langle 1 \rangle 2$ DEF *AcceptorSendAction*
 $\langle 2 \rangle 2$.CASE *Phase1b*(*lrn*, *bal*, *acc*)BY $\langle 2 \rangle 2$, $\langle 1 \rangle 0a$, $\langle 1 \rangle 0b$ DEF *AcceptorSendAction*, *Send*, *Phase1b*, *Safety*, *TypeOK*
 $\langle 2 \rangle 3$.CASE *Phase2av*(*lrn*, *bal*, *acc*, *val*)BY $\langle 2 \rangle 3$, $\langle 1 \rangle 0a$, $\langle 1 \rangle 0b$ DEF *AcceptorSendAction*, *Send*, *Phase2av*, *Safety*, *TypeOK*
 $\langle 2 \rangle 4$.CASE *Phase2b*(*lrn*, *bal*, *acc*, *val*)BY $\langle 2 \rangle 4$, $\langle 1 \rangle 0a$, $\langle 1 \rangle 0b$ DEF *AcceptorSendAction*, *Send*, *Phase2b*, *Safety*, *TypeOK*
 $\langle 2 \rangle 5$. QED BY $\langle 2 \rangle 2$, $\langle 2 \rangle 3$, $\langle 2 \rangle 4$
 $\langle 1 \rangle 3$.CASE *AcceptorReceiveAction*BY $\langle 1 \rangle 3$, $\langle 1 \rangle 0a$, $\langle 1 \rangle 0b$ DEF *AcceptorReceiveAction*, *Recv*, *TypeOK*, *Safety*
 $\langle 1 \rangle 4$.CASE *AcceptorDisconnectAction*BY $\langle 1 \rangle 4$ DEF *AcceptorDisconnectAction*, *Disconnect*, *Safety*
 $\langle 1 \rangle 5$.CASE *LearnerAction*
 $\langle 2 \rangle$ SUFFICES ASSUME NEW *lrn* \in *Learner*, NEW *bal* \in *Ballot*,
 \vee *LearnerDecide*(*lrn*, *bal*)
 \vee *LearnerRecv*(*lrn*)
 PROVE $V1 = V2$ BY $\langle 1 \rangle 5$ DEF *LearnerAction*
 $\langle 2 \rangle 1$.CASE *LearnerRecv*(*lrn*)BY $\langle 2 \rangle 1$ DEF *LearnerRecv*, *Safety*
 $\langle 2 \rangle 2$.CASE *LearnerDecide*(*lrn*, *bal*)
 $\langle 3 \rangle$ SUFFICES ASSUME NEW *val* \in *Value*,
ChosenIn(*lrn*, *bal*, *val*),
 $decision' = [decision \text{ EXCEPT } ![\langle lrn, bal \rangle] = decision[lrn, bal] \cup \{val\}]$,
 UNCHANGED $\langle msgs, maxBal, votesSent, 2avSent, received, connected, receivedByLearner \rangle$
 PROVE $V1 = V2$
 BY $\langle 2 \rangle 2$ DEF *LearnerDecide*
 $\langle 3 \rangle 0$.CASE $V1 = V2$ BY $\langle 3 \rangle 0$
 $\langle 3 \rangle 1$.CASE $V1 \neq V2$
 $\langle 4 \rangle 1$.CASE $val \neq V1 \wedge val \neq V2$ BY $\langle 4 \rangle 1$ DEF *Safety*, *TypeOK*
 $\langle 4 \rangle 2$.CASE $val = V1$
 $\langle 5 \rangle 0$. $V2 \in decision[L2, B2]$ BY $\langle 3 \rangle 1$, $\langle 4 \rangle 2$ DEF *TypeOK*
 $\langle 5 \rangle 1$. *ChosenIn*(*L2*, *B2*, *V2*)BY $\langle 5 \rangle 0$ DEF *DecisionSpec*
 $\langle 5 \rangle 2$.CASE $V1 \in decision[L1, B1]$ BY $\langle 5 \rangle 0$, $\langle 5 \rangle 2$ DEF *Safety*
 $\langle 5 \rangle 3$.CASE $V1 \notin decision[L1, B1]$
 $\langle 6 \rangle 1$. $lrn = L1 \wedge bal = B1$ BY $\langle 5 \rangle 3$, $\langle 4 \rangle 2$ DEF *TypeOK*
 $\langle 6 \rangle 2$. *ChosenIn*(*L1*, *B1*, *V1*)BY $\langle 6 \rangle 1$, $\langle 4 \rangle 2$
 $\langle 6 \rangle 3$. QED BY $\langle 5 \rangle 1$, $\langle 6 \rangle 2$, *ChosenSafe*
 $\langle 5 \rangle 4$. QED BY $\langle 5 \rangle 2$, $\langle 5 \rangle 3$
 $\langle 4 \rangle 3$.CASE $val = V2$
 $\langle 5 \rangle 0$. $V1 \in decision[L1, B1]$ BY $\langle 3 \rangle 1$, $\langle 4 \rangle 3$ DEF *TypeOK*
 $\langle 5 \rangle 1$. *ChosenIn*(*L1*, *B1*, *V1*)BY $\langle 5 \rangle 0$ DEF *DecisionSpec*
 $\langle 5 \rangle 2$.CASE $V2 \in decision[L2, B2]$ BY $\langle 5 \rangle 0$, $\langle 5 \rangle 2$ DEF *Safety*
 $\langle 5 \rangle 3$.CASE $V2 \notin decision[L2, B2]$
 $\langle 6 \rangle 1$. $lrn = L2 \wedge bal = B2$ BY $\langle 5 \rangle 3$, $\langle 4 \rangle 3$ DEF *TypeOK*
 $\langle 6 \rangle 2$. *ChosenIn*(*L2*, *B2*, *V2*)BY $\langle 6 \rangle 1$, $\langle 4 \rangle 3$
 $\langle 6 \rangle 10$. QED BY $\langle 5 \rangle 1$, $\langle 6 \rangle 2$, *ChosenSafe*
 $\langle 5 \rangle 4$. QED BY $\langle 5 \rangle 2$, $\langle 5 \rangle 3$
 $\langle 4 \rangle 4$. QED BY $\langle 4 \rangle 1$, $\langle 4 \rangle 2$, $\langle 4 \rangle 3$
 $\langle 3 \rangle 2$. QED BY $\langle 3 \rangle 0$, $\langle 3 \rangle 1$
 $\langle 2 \rangle 3$. QED BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$
 $\langle 1 \rangle 6$.CASE *FakeAcceptorAction*BY $\langle 1 \rangle 6$ DEF *FakeAcceptorAction*, *FakeSend*, *Send*, *Safety*

$\langle 1 \rangle 7$. QED BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, \langle 1 \rangle 4, \langle 1 \rangle 5, \langle 1 \rangle 6$ DEF *Next*

FullSafetyInvariant \triangleq
 \wedge *TypeOK*
 \wedge *MsgInv*
 \wedge *2avSentSpec1* \wedge *2avSentSpec2* \wedge *2avSentSpec3*
 \wedge *VotesSentSpec1* \wedge *VotesSentSpec2* \wedge *VotesSentSpec3* \wedge *VotesSentSpec4*
 \wedge *ReceivedSpec*
 \wedge *ReceivedByLearnerSpec*
 \wedge *ConnectedSpec*
 \wedge *DecisionSpec*
 \wedge *HeterogeneousSpec*
 \wedge *Safety*

LEMMA *TypeOKInit* \triangleq *Init* \Rightarrow *TypeOK*

PROOF BY DEF *Init*, *TypeOK*

LEMMA *MsgInvInit* \triangleq *Init* \Rightarrow *MsgInv*

PROOF BY DEF *Init*, *MsgInv*

LEMMA *2avSentSpec1Init* \triangleq *Init* \Rightarrow *2avSentSpec1*

PROOF BY DEF *Init*, *2avSentSpec1*

LEMMA *2avSentSpec2Init* \triangleq *Init* \Rightarrow *2avSentSpec2*

PROOF BY DEF *Init*, *2avSentSpec2*, *Proposed*

LEMMA *2avSentSpec3Init* \triangleq *Init* \Rightarrow *2avSentSpec3*

PROOF BY DEF *Init*, *2avSentSpec3*, *TypeOK*

LEMMA *VotesSentSpec1Init* \triangleq *Init* \Rightarrow *VotesSentSpec1*

PROOF BY DEF *Init*, *VotesSentSpec1*

LEMMA *VotesSentSpec2Init* \triangleq *Init* \Rightarrow *VotesSentSpec2*

PROOF BY DEF *Init*, *VotesSentSpec2*, *VotedFor*

LEMMA *VotesSentSpec3Init* \triangleq *Init* \Rightarrow *VotesSentSpec3*

PROOF BY DEF *Init*, *VotesSentSpec3*

LEMMA *VotesSentSpec4Init* \triangleq *Init* \Rightarrow *VotesSentSpec4*

PROOF BY DEF *Init*, *VotesSentSpec4*

LEMMA *ReceivedSpecInit* \triangleq *Init* \Rightarrow *ReceivedSpec*

PROOF BY *SafeAcceptorIsAcceptor* DEF *Init*, *ReceivedSpec*

LEMMA *ReceivedByLearnerSpecInit* \triangleq *Init* \Rightarrow *ReceivedByLearnerSpec*

PROOF BY DEF *Init*, *ReceivedByLearnerSpec*, *TypeOK*

LEMMA *ConnectedSpecInit* \triangleq *Init* \Rightarrow *ConnectedSpec*

PROOF BY DEF *Init*, *ConnectedSpec*

LEMMA $DecisionSpecInit \triangleq Init \Rightarrow DecisionSpec$

PROOF BY DEF $Init$, $DecisionSpec$

LEMMA $HeterogeneousSpecInit \triangleq Init \Rightarrow HeterogeneousSpec$

PROOF BY DEF $Init$, $HeterogeneousSpec$

LEMMA $SafetyInit \triangleq Init \Rightarrow Safety$

PROOF BY DEF $Init$, $Safety$

LEMMA $FullSafetyInvariantInit \triangleq Init \Rightarrow FullSafetyInvariant$

PROOF BY $TypeOKInit$, $MsgInvInit$,
 $2avSentSpec1Init$, $2avSentSpec2Init$, $2avSentSpec3Init$,
 $VotesSentSpec1Init$, $VotesSentSpec2Init$, $VotesSentSpec3Init$, $VotesSentSpec4Init$,
 $ReceivedSpecInit$, $ReceivedByLearnerSpecInit$, $ConnectedSpecInit$, $DecisionSpecInit$,
 $HeterogeneousSpecInit$, $SafetyInit$
 DEF $FullSafetyInvariant$

LEMMA $FullSafetyInvariantNext \triangleq FullSafetyInvariant \wedge [Next]_{vars} \Rightarrow FullSafetyInvariant'$

PROOF

$\langle 1 \rangle$ SUFFICES ASSUME $FullSafetyInvariant$, $[Next]_{vars}$ PROVE $FullSafetyInvariant'$ OBVIOUS

$\langle 1 \rangle 1$. CASE $Next$ BY $\langle 1 \rangle 1$,

$TypeOKInvariant$, $MsgInvInvariant$,
 $2avSentSpec1Invariant$, $2avSentSpec2Invariant$, $2avSentSpec3Invariant$,
 $VotesSentSpec1Invariant$, $VotesSentSpec2Invariant$, $VotesSentSpec3Invariant$, $VotesSentSpec4Invariant$,
 $ReceivedSpecInvariant$, $ReceivedByLearnerSpecInvariant$, $ConnectedSpecInvariant$, $DecisionSpecInvariant$,
 $HeterogeneousSpecInvariant$, $SafetyStep$

DEF $FullSafetyInvariant$

$\langle 1 \rangle 2$. CASE $vars = vars'$ BY $\langle 1 \rangle 2$ DEF $vars$, $FullSafetyInvariant$, $TypeOK$, $MsgInv$,

$2avSentSpec1$, $2avSentSpec2$, $2avSentSpec3$,
 $VotesSentSpec1$, $VotesSentSpec2$, $VotesSentSpec3$, $VotesSentSpec4$,
 $ReceivedSpec$, $ReceivedByLearnerSpec$, $ConnectedSpec$, $DecisionSpec$,
 $MsgInv1b$, $MsgInv2av$, $MsgInv2b$,
 $Safety$

$\langle 1 \rangle 3$. QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$

THEOREM $SafetyResult \triangleq Spec \Rightarrow \Box Safety$

PROOF BY PTL , $FullSafetyInvariantInit$, $FullSafetyInvariantNext$ DEF $Spec$, $FullSafetyInvariant$