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 $A < B \lor B < A \lor A = B$
PROOF BY DEF *Ballot*

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

$None \triangleq$ CHOOSE $v : v \notin Value$

CONSTANTS *Acceptor*,
SafeAcceptor,
FakeAcceptor,
ByzQuorum,
Learner

ASSUME $SafeAcceptorAssumption \triangleq$
$\land SafeAcceptor \cap FakeAcceptor = \{\}$
$\land SafeAcceptor \cup FakeAcceptor = Acceptor$

LEMMA $SafeAcceptorIsAcceptor \triangleq SafeAcceptor \subseteq Acceptor$
PROOF BY *SafeAcceptorAssumption*

LEMMA $FakeAcceptorIsAcceptor \triangleq FakeAcceptor \subseteq Acceptor$
PROOF BY *SafeAcceptorAssumption*

ASSUME $BQAssumption \triangleq \forall Q \in ByzQuorum : Q \subseteq Acceptor$

ASSUME $BallotAssumption \triangleq$

$\land\ (Ballot \cup \{-1\}) \cap Acceptor = \{\}$
$\land\ (Ballot \cup \{-1\}) \cap ByzQuorum = \{\}$
$\land\ (Ballot \cup \{-1\}) \cap Learner = \{\}$

Learner graph

CONSTANT $TrustLive$
ASSUME $TrustLiveAssumption \triangleq TrustLive \in \text{SUBSET } [lr : Learner,\ q : ByzQuorum]$

CONSTANT $TrustSafe$
ASSUME $TrustSafeAssumption \triangleq TrustSafe \in \text{SUBSET } [from : Learner,\ to : Learner,\ q : ByzQuorum]$

ASSUME $LearnerGraphAssumption \triangleq$

symmetry
$\land \forall E \in TrustSafe :$
$\quad [from \mapsto E.to,\ to \mapsto E.from,\ q \mapsto E.q] \in TrustSafe$

transitivity
$\land \forall E1,\ E2 \in TrustSafe :$
$\quad E1.q = E2.q \land E1.to = E2.from \Rightarrow$
$\quad [from \mapsto E1.from,\ to \mapsto E2.to,\ q \mapsto E1.q] \in TrustSafe$

closure
$\land \forall E \in TrustSafe : \forall Q \in ByzQuorum :$
$\quad E.q \subseteq Q \Rightarrow$
$\quad [from \mapsto E.from,\ to \mapsto E.to,\ q \mapsto Q] \in TrustSafe$

validity
$\land \forall E \in TrustSafe : \forall Q1,\ Q2 \in ByzQuorum :$
$\quad [lr \mapsto E.from,\ q \mapsto Q1] \in TrustLive \land$
$\quad [lr \mapsto E.to,\ q \mapsto Q2] \in TrustLive \Rightarrow$
$\quad \exists N \in E.q : N \in Q1 \land N \in Q2$

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

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 $L1 \in Learner$, NEW $L2 \in Learner$,
NEW $Q1 \in ByzQuorum$, NEW $Q2 \in ByzQuorum$,
$\langle L1, L2 \rangle \in Ent$,
$[lr \mapsto L1, q \mapsto Q1] \in TrustLive$,
$[lr \mapsto L2, q \mapsto Q2] \in TrustLive$
PROVE $\exists N \in SafeAcceptor : N \in Q1 \land N \in Q2$
PROOF BY $EntanglementAssumption$, $LearnerGraphAssumption$

Messages

$$
\begin{aligned}
&Message \triangleq \\
&\quad [type : \{\text{“1a”}\}, lr : Learner, bal : Ballot] \cup \\
&\quad [\\
&\qquad type : \{\text{“1b”}\}, \\
&\qquad lr \quad : Learner, \\
&\qquad acc \ : Acceptor, \\
&\qquad bal \ : Ballot, \\
&\qquad votes : \text{SUBSET } [lr : Learner, bal : Ballot, val : Value], \\
&\qquad proposals : \text{SUBSET } [lr : Learner, bal : Ballot, val : Value] \\
&\quad] \cup \\
&\quad [type : \{\text{“1c”}\}, lr : Learner, bal : Ballot, val : Value] \cup \\
&\quad [type : \{\text{“2av”}\}, lr : Learner, acc : Acceptor, bal : Ballot, val : Value] \cup \\
&\quad [type : \{\text{“2b”}\}, lr : Learner, acc : Acceptor, bal : Ballot, val : Value]
\end{aligned}
$$

Algorithm specification

$$
\begin{aligned}
&\text{VARIABLES } maxBal, \\
&\qquad\qquad votesSent, \\
&\qquad\qquad 2avSent, \\
&\qquad\qquad received, \\
&\qquad\qquad connected, \\
&\qquad\qquad receivedByLearner, \\
&\qquad\qquad decision, \\
&\qquad\qquad msgs
\end{aligned}
$$

$$
\begin{aligned}
&InitializedBallot(lr, bal) \triangleq \\
&\quad \exists\, m \in msgs : m.type = \text{“1a”} \land m.lr = lr \land m.bal = bal
\end{aligned}
$$

$$
\begin{aligned}
&AnnouncedValue(lr, bal, val) \triangleq \\
&\quad \exists\, m \in msgs : m.type = \text{“1c”} \land m.bal = bal \land m.val = val
\end{aligned}
$$

$$
\begin{aligned}
&ChosenIn(lr, bal, v) \triangleq \\
&\quad \exists\, Q \in ByzQuorum : \\
&\qquad \land [lr \mapsto lr, q \mapsto Q] \in TrustLive \\
&\qquad \land \forall\, aa \in Q :
\end{aligned}
$$

$$
\begin{array}{l}
\quad \exists\, m \in \{mm \in receivedByLearner[lr] : mm.bal = bal\} : \\
\qquad \land\, m.val = v \\
\qquad \land\, m.acc = aa
\end{array}
$$

$$
\begin{array}{l}
KnowsSafeAt1(l,\, ac,\, b,\, v) \triangleq \\
\quad \text{LET } S \triangleq \{mm \in received[ac] : mm.type = \text{“1b”} \land mm.lr = l \land mm.bal = b\} \\
\quad \text{IN } \exists\, BQ \in ByzQuorum : \\
\qquad \land\, [lr \mapsto l,\, q \mapsto BQ] \in TrustLive \\
\qquad \land\, \forall\, a \in BQ : \\
\qquad\quad \exists\, m \in S\ : \\
\qquad\qquad \land\, m.acc = a \\
\qquad\qquad \land\, \forall\, p \in \{pp \in m.votes : \langle pp.lr,\, l\rangle \in connected[ac]\} : \\
\qquad\qquad\quad b \leq p.bal
\end{array}
$$

$$
\begin{array}{l}
KnowsSafeAt2(l,\, ac,\, b,\, v) \triangleq \\
\quad \text{LET } S \triangleq \{mm \in received[ac] : mm.type = \text{“1b”} \land mm.lr = l \land mm.bal = b\} \\
\quad \text{IN } \exists\, c \in Ballot : \\
\qquad \land\, c < b \\
\qquad \land\, \exists\, BQ \in ByzQuorum : \\
\qquad\quad \land\, [lr \mapsto l,\, q \mapsto BQ] \in TrustLive \\
\qquad\quad \land\, \forall\, a \in BQ : \\
\qquad\qquad \exists\, m \in S\ : \\
\qquad\qquad\quad \land\, m.acc = a \\
\qquad\qquad\quad \land\, \forall\, p \in \{pp \in m.votes : \langle pp.lr,\, l\rangle \in connected[ac]\} : \\
\qquad\qquad\qquad \land\, p.bal \leq c \\
\qquad\qquad\qquad \land\, (p.bal = c) \Rightarrow (p.val = v) \\
\qquad \land\, \exists\, WQ \in ByzQuorum : \\
\qquad\quad \land\, [lr \mapsto l,\, q \mapsto WQ] \in TrustLive \\
\qquad\quad \land\, \forall\, a \in WQ : \\
\qquad\qquad \exists\, m \in S\ : \\
\qquad\qquad\quad \land\, m.acc = a \\
\qquad\qquad\quad \land\, \exists\, p \in m.proposals : \\
\qquad\qquad\qquad \land\, p.lr = l \\
\qquad\qquad\qquad \land\, p.bal = c \\
\qquad\qquad\qquad \land\, p.val = v
\end{array}
$$

$$
\begin{array}{l}
KnowsSafeAt(l,\, ac,\, b,\, v) \triangleq \\
\quad \lor\, KnowsSafeAt1(l,\, ac,\, b,\, v) \\
\quad \lor\, KnowsSafeAt2(l,\, ac,\, b,\, v)
\end{array}
$$

$$
vars \triangleq \langle maxBal,\, votesSent,\, 2avSent,\, received,\, connected,\, receivedByLearner,\, decision,\, msgs\rangle
$$

$$
\begin{array}{l}
TypeOK \triangleq \\
\quad \land\quad msgs \in \text{SUBSET } Message \\
\quad \land\quad maxBal \in [Learner \times Acceptor \to Ballot] \\
\quad \land\quad votesSent \in [Acceptor \to \text{SUBSET } [lr : Learner,\, bal : Ballot,\, val : Value]]
\end{array}
$$

$\land \quad 2avSent \quad \in [Acceptor \to \text{SUBSET } [lr : Learner,\ bal : Ballot,\ val : Value]]$
$\land \quad connected \in [Acceptor \to \text{SUBSET } (Learner \times Learner)]$
$\land \quad received \quad \in [Acceptor \to \text{SUBSET } Message]$
$\land \quad receivedByLearner \in [Learner \to \text{SUBSET } Message]$
$\land \quad decision \in [Learner \times Ballot \to \text{SUBSET } Value]$

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

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

$Phase1a(l,\ b) \triangleq$
$\land \quad Send([type \mapsto \text{“}\mathsf{1a}\text{”},\ lr \mapsto l,\ bal \mapsto b])$
$\land \quad \text{UNCHANGED } \langle maxBal,\ votesSent,\ 2avSent,\ received,\ connected,\ receivedByLearner,\ decision\rangle$

$Phase1c(l,\ b,\ v) \triangleq$
$\land \quad Send([type \mapsto \text{“}\mathsf{1c}\text{”},\ lr \mapsto l,\ bal \mapsto b,\ val \mapsto v])$
$\land \quad \text{UNCHANGED } \langle maxBal,\ votesSent,\ 2avSent,\ received,\ connected,\ receivedByLearner,\ decision\rangle$

$MaxVote(a,\ b,\ vote) \triangleq$
$\land \quad vote.bal < b$
$\land \quad \forall\, other \ \in votesSent[a] :$
$\quad\quad other.lr = vote.lr \land other.bal < b \Rightarrow$
$\quad\quad other.bal \leq vote.bal$

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

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

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

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

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

$FakeSend(a) \triangleq$
$\quad \land\ \exists\, m \in \{mm \in Message :$
$\qquad\qquad \land\ mm.acc = a$
$\qquad\qquad \land\ \lor\ mm.type = \text{“1b”}$
$\qquad\qquad\quad \lor\ mm.type = \text{“2av”}$
$\qquad\qquad\quad \lor\ mm.type = \text{“2b”}\} :$
$\qquad Send(m)$
$\quad \land\ \text{UNCHANGED } \langle maxBal,\ votesSent,\ 2avSent,\ received,\ connected,\ receivedByLearner,\ decision\rangle$

$LearnerDecide(l,\ b) \triangleq$
$\quad \land\ \exists\, v \in \{vv \in Value : ChosenIn(l,\ b,\ vv)\} :$

$decision' = [decision \text{ EXCEPT } ![l,\ b] = decision[l,\ b] \cup \{v\}]$
$\land$ UNCHANGED $\langle msgs,\ maxBal,\ votesSent,\ 2avSent,\ received,\ connected,\ receivedByLearner\rangle$

$LearnerRecv(l) \triangleq$
$\land \exists\, m \in \{mm \in msgs : mm.type = \text{“2b”} \land mm.lr = l\} :$
$receivedByLearner' =$
$[receivedByLearner \text{ EXCEPT } ![l] = receivedByLearner[l] \cup \{m\}]$
$\land$ UNCHANGED $\langle msgs,\ maxBal,\ votesSent,\ 2avSent,\ received,\ connected,\ decision\rangle$

$ProposerAction \triangleq$
$\exists\, lrn \in Learner : \exists\, proposer \in Ballot :$
$\lor Phase1a(lrn,\ proposer)$
$\lor \exists\, v \in Value : Phase1c(lrn,\ proposer,\ v)$

$AcceptorSendAction \triangleq$
$\exists\, lrn \in Learner : \exists\, bal \in Ballot : \exists\, acc \in SafeAcceptor : \exists\, val \in Value :$
$\lor Phase1b(lrn,\ bal,\ acc)$
$\lor Phase2av(lrn,\ bal,\ acc,\ val)$
$\lor Phase2b(lrn,\ bal,\ acc,\ val)$

$AcceptorReceiveAction \triangleq$
$\exists\, lrn \in Learner : \exists\, acc \in Acceptor : Recv(lrn,\ acc)$

$AcceptorDisconnectAction \triangleq$
$\exists\, acc \in SafeAcceptor : Disconnect(acc)$

$LearnerAction \triangleq$
$\exists\, lrn \in Learner :$
$\lor \exists\, bal \in Ballot : LearnerDecide(lrn,\ bal)$
$\lor LearnerRecv(lrn)$

$FakeAcceptorAction \triangleq \exists\, a \in FakeAcceptor : FakeSend(a)$

$Next \triangleq$
$\lor ProposerAction$
$\lor AcceptorSendAction$
$\lor AcceptorReceiveAction$
$\lor AcceptorDisconnectAction$
$\lor LearnerAction$
$\lor FakeAcceptorAction$

$Spec \triangleq Init \land \Box[Next]_{vars}$

$VotedFor(lr,\ acc,\ bal,\ val) \triangleq$
$\exists\, m \in msgs :$
$\land m.type = \text{“2b”}$

$\land\ m.lr = lr$
$\land\ m.acc = acc$
$\land\ m.bal = bal$
$\land\ m.val = val$

$Proposed(lr,\ acc,\ bal,\ val) \triangleq$
$\exists\, m \in msgs :$
$\land\ m.type =$ "2av"
$\land\ m.lr = lr$
$\land\ m.acc = acc$
$\land\ m.bal = bal$
$\land\ m.val = val$

$LeftBallot(lr,\ acc,\ bal) \triangleq$
$\exists\, m \in msgs :$
$\land\ m.type =$ "1b"
$\land\ m.lr = lr$
$\land\ m.acc = acc$
$\land\ bal < m.bal$

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

$ReceivedByLearnerSpec \triangleq$
$\land\ receivedByLearner \in [Learner \to \text{SUBSET}\ \{mm \in msgs : mm.type =$ "2b"$\}]$
$\land \forall\, L \in Learner : \forall\, mm \in Message :$
$mm \in receivedByLearner[L] \Rightarrow mm.lr = L$

$VotesSentSpec1 \triangleq$
$\forall\, A \in SafeAcceptor : \forall\, vote \in votesSent[A] : VotedFor(vote.lr,\ A,\ vote.bal,\ vote.val)$

$VotesSentSpec2 \triangleq$
$\forall\, L \in Learner : \forall\, A \in SafeAcceptor : \forall\, B \in Ballot : \forall\, V \in Value :$
$VotedFor(L,\ A,\ B,\ V) \Rightarrow [lr \mapsto L,\ bal \mapsto B,\ val \mapsto V] \in votesSent[A]$

$VotesSentSpec3 \triangleq$
$\forall\, A \in SafeAcceptor : \forall\, B \in Ballot : \forall\, vote \in votesSent[A] :$
$vote.bal < B \Rightarrow$
$\exists\, P \in votesSent[A] :$
$MaxVote(A,\ B,\ P) \land P.lr = vote.lr \land vote.bal \leq P.bal$

$VotesSentSpec4 \triangleq$
$\forall\, A \in SafeAcceptor : \forall\, vote1,\ vote2 \in votesSent[A] :$
$\langle vote1.lr,\ vote2.lr\rangle \in Ent \land$
$vote1.bal = vote2.bal \Rightarrow vote1.val = vote2.val$

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

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

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

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

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

$$
\begin{array}{l}
MsgInv1b(m) \triangleq \\
\quad \land\, m.bal \leq maxBal[m.lr,\, m.acc] \\
\quad \land\, m.votes = \{p \in votesSent[m.acc] : MaxVote(m.acc,\, m.bal,\, p)\} \\
\quad \land\, m.proposals = \{p \in 2avSent[m.acc] : p.bal < m.bal \land p.lr = m.lr\}
\end{array}
$$

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

$$
\begin{array}{l}
MsgInv2b(m) \triangleq \\
\quad \land\, [lr \mapsto m.lr,\, bal \mapsto m.bal,\, val \mapsto m.val] \in votesSent[m.acc] \\
\quad \land\, \exists\, Q \in ByzQuorum : \\
\qquad \land\, [lr \mapsto m.lr,\, q \mapsto Q] \in TrustLive \\
\qquad \land\, \forall\, ba \in Q : \\
\qquad\quad \exists\, m2av \in received[m.acc] : \\
\qquad\qquad \land\, m2av.type = \text{“2av”} \\
\qquad\qquad \land\, m2av.lr = m.lr \\
\qquad\qquad \land\, m2av.acc = ba \\
\qquad\qquad \land\, m2av.bal\ = m.bal
\end{array}
$$

$\land\ m2av.val = m.val$

$MsgInv \triangleq \forall\, m \in msgs : m.acc \in SafeAcceptor \Rightarrow$
$\land\ (m.type = \text{“1b”}) \Rightarrow MsgInv1b(m)$
$\land\ (m.type = \text{“2av”}) \Rightarrow MsgInv2av(m)$
$\land\ (m.type = \text{“2b”}) \Rightarrow MsgInv2b(m)$

LEMMA $MessageType \triangleq$
ASSUME NEW $m \in Message$
PROVE $\land\ m.lr \in Learner$
$\land\ m.bal \in Ballot$
$\land\ (m.type = \text{“1b”} \lor m.type = \text{“2av”} \lor m.type = \text{“2b”}) \Rightarrow m.acc \in Acceptor$
$\land\ (m.type = \text{“1c”} \lor m.type = \text{“2av”} \lor m.type = \text{“2b”}) \Rightarrow m.val \in Value$
$\land\ (m.type = \text{“1b”}) \Rightarrow$
$\land\ m.votes \in \text{SUBSET}\ [lr : Learner,\ bal : Ballot,\ val : Value]$
$\land\ m.proposals \in \text{SUBSET}\ [lr : Learner,\ bal : Ballot,\ val : Value]$
PROOF BY DEF $Message$

LEMMA $TypeOKInvariant \triangleq TypeOK \land Next \Rightarrow TypeOK'$
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 $val \in Value$,
$\lor\ Phase1b(lrn,\ bal,\ acc)$
$\lor\ Phase2av(lrn,\ bal,\ acc,\ val)$
$\lor\ Phase2b(lrn,\ bal,\ acc,\ val)$
PROVE $TypeOK'$
BY $\langle 1\rangle$2, $SafeAcceptorIsAcceptor$ DEF $AcceptorSendAction$
$\langle 2\rangle$1.CASE $Phase1b(lrn,\ bal,\ acc)$
$\langle 3\rangle$1. $(votesSent \in [Acceptor \to \text{SUBSET}\ [lr : Learner,\ bal : Ballot,\ val : Value]])'$
BY $\langle 2\rangle$1 DEF $Phase1b$, $Phase2av$, $Phase2b$, $Send$, $TypeOK$, $Message$
$\langle 3\rangle$2. $(2avSent \in [Acceptor \to \text{SUBSET}\ [lr : Learner,\ bal : Ballot,\ val : Value]])'$
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
$[type \mapsto \text{“1b”},\ lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,$
$votes \mapsto \{vote \in votesSent[acc] : MaxVote(acc,\ bal,\ vote)\},$
$proposals \mapsto \{p \in 2avSent[acc] : p.bal < bal \land p.lr = lrn\}] \in Message$
BY $\langle 2\rangle$1 DEF $Phase1b$, $Send$, $TypeOK$

⟨4⟩1. $\{vote \in votesSent[acc] : MaxVote(acc, bal, vote)\}$
$\in$ SUBSET $[lr : Learner,\ bal : Ballot,\ val : Value]$
BY DEF *TypeOK*
⟨4⟩2. $\{p \in 2avSent[acc] : p.bal < bal \wedge p.lr = lrn\} \in$ SUBSET $[lr : Learner,\ bal : Ballot,\ val : Value]$
BY DEF *TypeOK*
⟨4⟩3. QED BY ⟨4⟩1, ⟨4⟩2 DEF *Message*, *TypeOK*
⟨3⟩4. QED BY ⟨2⟩1, ⟨3⟩1, ⟨3⟩2, ⟨3⟩3 DEF *Phase*1*b*, *TypeOK*, *Send*
⟨2⟩2.CASE *Phase*2*av*$(lrn,\ bal,\ acc,\ val)$
⟨3⟩2. $msgs' \in$ SUBSET *Message*
⟨4⟩0. $[type \mapsto$ "2av", $lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val] \in Message$
BY *SafeAcceptorIsAcceptor* DEF *Message*
⟨4⟩1. QED BY ⟨2⟩2, ⟨4⟩0, *SafeAcceptorIsAcceptor* DEF *Phase*2*av*, *Send*, *TypeOK*, *Message*
⟨3⟩4. $(2avSent \in [Acceptor \to$ SUBSET $[lr : Learner,\ bal : Ballot,\ val : Value]])'$
⟨4⟩0. $[lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val] \in [lr : Learner,\ bal : Ballot,\ val : Value]$
BY DEF *TypeOK*
⟨4⟩1. QED BY ⟨2⟩2, ⟨1⟩2, ⟨4⟩0, *SafeAcceptorIsAcceptor* DEF *Phase*2*av*, *Send*, *TypeOK*, *Message*
⟨3⟩5. QED BY ⟨2⟩2, ⟨3⟩2, ⟨3⟩4 DEF *Phase*2*av*, *Send*, *TypeOK*
⟨2⟩3.CASE *Phase*2*b*$(lrn,\ bal,\ acc,\ val)$
⟨3⟩1. $val \in Value$OBVIOUS
⟨3⟩2. $msgs' \in$ SUBSET *Message*
⟨4⟩0. $[type \mapsto$ "2b", $lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val] \in Message$
BY *SafeAcceptorIsAcceptor* DEF *Message*
⟨4⟩1. QED BY ⟨4⟩0, ⟨2⟩3 DEF *Phase*2*b*, *Message*, *Send*, *TypeOK*
⟨3⟩3. $votesSent' \in [Acceptor \to$ SUBSET $[lr : Learner,\ bal : Ballot,\ val : Value]]$
⟨4⟩0. $[lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val] \in [lr : Learner,\ bal : Ballot,\ val : Value]$BY ⟨3⟩1
⟨4⟩1 QED BY ⟨2⟩3, ⟨1⟩2, ⟨4⟩0 DEF *Phase*2*b*, *TypeOK*
⟨3⟩5. QED BY ⟨2⟩3, ⟨1⟩2, ⟨3⟩1, ⟨3⟩2, ⟨3⟩3 DEF *Phase*2*b*, *Send*, *TypeOK*
⟨2⟩4. QED BY ⟨1⟩2, ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3.CASE *AcceptorReceiveAction*
⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $acc \in Acceptor$,
NEW $m \in msgs$,
$received' = [received$ EXCEPT $![acc] = received[acc] \cup \{m\}]$,
UNCHANGED $\langle msgs,\ maxBal,\ 2avSent,\ votesSent,\ connected,$
$receivedByLearner,\ decision\rangle$
PROVE *TypeOK*′
BY *SafeAcceptorIsAcceptor*, ⟨1⟩3 DEF *AcceptorReceiveAction*, *Recv*
⟨2⟩7. QED BY ⟨1⟩3 DEF *AcceptorReceiveAction*, *Recv*, *TypeOK*
⟨1⟩4.CASE *AcceptorDisconnectAction*BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*, *TypeOK*, *Message*
⟨1⟩5.CASE *LearnerAction*
⟨2⟩1. ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$,
LearnerDecide$(lrn,\ bal)$
PROVE *TypeOK*′
BY ⟨2⟩1 DEF *LearnerDecide*, *TypeOK*
⟨2⟩2. 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 $ReceivedSpecInvariant \triangleq TypeOK \land ReceivedSpec \land Next \Rightarrow ReceivedSpec'$
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 $ProposerAction$, $Phase1a$, $Phase1c$, $ReceivedSpec$, $Send$, $Next$, Typ
$\langle 1\rangle 2$.CASE $AcceptorSendAction$
$\langle 2\rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in Acceptor$,
NEW $val \in Value$,
$\lor Phase1b(lrn, bal, acc)$
$\lor Phase2av(lrn, bal, acc, val)$
$\lor Phase2b(lrn, bal, acc, val)$
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 $\langle msgs, maxBal, 2avSent, votesSent, connected, receivedByLearner, decis$
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 $a \in Acceptor$, $FakeSend(a)$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 $ReceivedByLearnerSpecInvariant \triangleq$
$TypeOK \land ReceivedByLearnerSpec \land Next \Rightarrow ReceivedByLearnerSpec'$
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 $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in Acceptor$,
NEW $val \in Value$,
$\lor Phase1b(lrn, bal, acc)$
$\lor Phase2av(lrn, bal, acc, val)$
$\lor Phase2b(lrn, bal, acc, val)$
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 $Send([type \mapsto \text{“2b”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val])$
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 $Phase2b$
$\langle 3\rangle 3$. $(\forall L \in Learner : \forall mm \in Message : mm \in receivedByLearner[L] \Rightarrow mm.lr = L)'$
BY $\langle 3\rangle 1$ DEF *ReceivedByLearnerSpec*, *TypeOK*
$\langle 3\rangle 4$. $(receivedByLearner \in [Learner \to \text{SUBSET } \{mm \in msgs : mm.type = \text{“2b”}\}])'$
BY $\langle 3\rangle 0$, $\langle 3\rangle 1$, *MessageType* DEF *ReceivedByLearnerSpec*, *Send*, *TypeOK*

⟨3⟩5. QED BY ⟨3⟩3, ⟨3⟩4 DEF *ReceivedByLearnerSpec*
⟨2⟩4. QED BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3.CASE *AcceptorReceiveAction*
BY ⟨1⟩3 DEF *AcceptorReceiveAction*, *Recv*, *ReceivedByLearnerSpec*, *TypeOK*, *Next*
⟨1⟩4.CASE *AcceptorDisconnectAction*
BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*, *ReceivedByLearnerSpec*, *TypeOK*, *Next*
⟨1⟩5.CASE *LearnerAction*
⟨2⟩1. ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$, $LearnerDecide(lrn, bal)$
PROVE $ReceivedByLearnerSpec'$
BY ⟨2⟩1 DEF *LearnerDecide*, *ReceivedByLearnerSpec*, *TypeOK*, *Next*
⟨2⟩2. ASSUME NEW $lrn \in Learner$, $LearnerRecv(lrn)$
PROVE $ReceivedByLearnerSpec'$
⟨3⟩ SUFFICES ASSUME NEW $m \in \{mm \in msgs : mm.type = \text{“2b”} \land mm.lr = lrn\}$,
$receivedByLearner' =$
$[receivedByLearner \text{ EXCEPT } ![lrn] = receivedByLearner[lrn] \cup \{m\}]$
PROVE $ReceivedByLearnerSpec'$
BY ⟨2⟩2 DEF *LearnerRecv*
⟨3⟩1. UNCHANGED $\langle msgs \rangle$BY ⟨2⟩2 DEF *LearnerAction*, *LearnerRecv*
⟨3⟩5. QED BY ⟨2⟩2, ⟨3⟩1 DEF *ReceivedByLearnerSpec*
⟨2⟩3. QED BY ⟨1⟩5, ⟨2⟩1, ⟨2⟩2 DEF *LearnerAction*
⟨1⟩6.CASE *FakeAcceptorAction*
⟨2⟩1. SUFFICES ASSUME NEW $a \in Acceptor$, $FakeSend(a)$PROVE $ReceivedByLearnerSpec'$
BY ⟨1⟩6, *FakeAcceptorIsAcceptor* DEF *FakeAcceptorAction*
⟨2⟩2. QED BY ⟨2⟩1 DEF *FakeSend*, *Send*, *TypeOK*, *ReceivedByLearnerSpec*
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

LEMMA $MaxBalMonotone \triangleq$
$TypeOK \land Next \Rightarrow \forall\, l \in Learner : \forall\, a \in SafeAcceptor : maxBal[l,\, a] \leq maxBal'[l,\, a]$
PROOF
⟨1⟩ SUFFICES ASSUME $TypeOK$, $Next$, NEW CONSTANT $l \in Learner$, NEW CONSTANT $a \in SafeAcceptor$
PROVE $maxBal[l,\, a] \leq maxBal'[l,\, a]$
OBVIOUS
⟨1⟩1.CASE *ProposerAction*
BY ⟨1⟩1, *SafeAcceptorIsAcceptor* DEF *ProposerAction*, $Phase1a$, $Phase1c$, *Send*, *TypeOK*, *Ballot*
⟨1⟩2.CASE *AcceptorSendAction*
⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in Acceptor$,
NEW $val \in Value$,
$\lor\, Phase1b(lrn,\, bal,\, acc)$
$\lor\, Phase2av(lrn,\, bal,\, acc,\, val)$
$\lor\, Phase2b(lrn,\, bal,\, acc,\, val)$
PROVE $maxBal[l,\, a] \leq (maxBal')[l,\, a]$
BY ⟨1⟩2, *SafeAcceptorIsAcceptor* DEF *AcceptorSendAction*
⟨2⟩1.CASE $Phase1b(lrn,\, bal,\, acc)$

⟨3⟩1.CASE $\langle l,\ a\rangle = \langle lrn,\ acc\rangle$BY ⟨2⟩1, ⟨3⟩1 DEF *Phase*1*b*, *TypeOK*, *Ballot*
⟨3⟩2.CASE $\langle l,\ a\rangle \neq \langle lrn,\ acc\rangle$BY ⟨2⟩1, ⟨3⟩2, *SafeAcceptorIsAcceptor* DEF *Phase*1*b*, *TypeOK*, *Ballot*
⟨3⟩3. QED BY ⟨3⟩1, ⟨3⟩2
⟨2⟩2.CASE *Phase*2*av*(*lrn*, *bal*, *acc*, *val*)
⟨3⟩1. UNCHANGED *maxBal*BY ⟨2⟩2 DEF *Phase*2*av*
⟨3⟩2. QED BY ⟨3⟩1, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *Ballot*
⟨2⟩3.CASE *Phase*2*b*(*lrn*, *bal*, *acc*, *val*)
⟨3⟩1. UNCHANGED *maxBal*BY ⟨2⟩3 DEF *Phase*2*b*
⟨3⟩2. QED BY ⟨3⟩1, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *Ballot*
⟨2⟩4. QED BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3.CASE *AcceptorReceiveAction*
⟨2⟩1. UNCHANGED *maxBal*BY ⟨1⟩3 DEF *AcceptorReceiveAction*, *Recv*
⟨2⟩2. QED BY ⟨2⟩1, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *Ballot*
⟨1⟩4.CASE *AcceptorDisconnectAction*
⟨2⟩1. UNCHANGED *maxBal*BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*
⟨2⟩2. QED BY ⟨2⟩1, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *Ballot*
⟨1⟩5.CASE *LearnerAction*
⟨2⟩1. UNCHANGED *maxBal*BY ⟨1⟩5 DEF *LearnerAction*, *LearnerDecide*, *LearnerRecv*
⟨2⟩2. QED BY ⟨2⟩1, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *Ballot*
⟨1⟩6.CASE *FakeAcceptorAction*
⟨2⟩1. UNCHANGED *maxBal*BY ⟨1⟩6 DEF *FakeAcceptorAction*, *FakeSend*
⟨2⟩2. QED BY ⟨2⟩1, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *Ballot*
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

LEMMA $2avSentMonotone \triangleq TypeOK \land Next \Rightarrow \forall\, A \in SafeAcceptor : 2avSent[A] \subseteq 2avSent'[A]$
PROOF
⟨1⟩ SUFFICES ASSUME *TypeOK*, *Next*, NEW $A \in SafeAcceptor$PROVE $2avSent[A] \subseteq 2avSent[A]'$OBVIOUS
⟨1⟩0a. *TypeOK*OBVIOUS
⟨1⟩0b. *TypeOK*′BY *TypeOKInvariant*
⟨1⟩1.CASE *ProposerAction*BY ⟨1⟩1 DEF *ProposerAction*, *Phase*1*a*, *Phase*1*c*, *Send*
⟨1⟩2.CASE *AcceptorSendAction*
⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in Acceptor$,
NEW $val \in Value$,
$\lor$ *Phase*1*b*(*lrn*, *bal*, *acc*)
$\lor$ *Phase*2*av*(*lrn*, *bal*, *acc*, *val*)
$\lor$ *Phase*2*b*(*lrn*, *bal*, *acc*, *val*)
PROVE $2avSent[A] \subseteq 2avSent[A]'$
BY ⟨1⟩2, *SafeAcceptorIsAcceptor* DEF *AcceptorSendAction*
⟨2⟩1. QED BY ⟨1⟩0b, *SafeAcceptorIsAcceptor* DEF *AcceptorSendAction*, *Phase*1*b*, *Phase*2*av*, *Phase*2*b*, *Send*,
⟨1⟩3.CASE *AcceptorReceiveAction*BY ⟨1⟩3 DEF *AcceptorReceiveAction*, *Recv*
⟨1⟩4.CASE *AcceptorDisconnectAction*BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*
⟨1⟩5.CASE *LearnerAction*BY ⟨1⟩5 DEF *LearnerAction*, *LearnerDecide*, *LearnerRecv*
⟨1⟩6.CASE *FakeAcceptorAction*BY ⟨1⟩6 DEF *FakeAcceptorAction*, *FakeSend*, *Send*

⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

LEMMA *ReceivedMonotone* $\triangleq$
$TypeOK \land Next \Rightarrow \forall A \in SafeAcceptor : received[A] \subseteq received'[A]$
PROOF
⟨1⟩ SUFFICES ASSUME *TypeOK*, *Next*, NEW $A \in SafeAcceptor$
PROVE $received[A] \subseteq received'[A]$OBVIOUS
⟨1⟩0a. *TypeOK* OBVIOUS
⟨1⟩0b. *TypeOK*′ BY *TypeOKInvariant*
⟨1⟩1.CASE *ProposerAction*BY ⟨1⟩1 DEF *ProposerAction*, *Phase*1*a*, *Phase*1*c*, *Send*
⟨1⟩2.CASE *AcceptorSendAction*BY ⟨1⟩2 DEF *AcceptorSendAction*, *Send*, *Phase*1*b*, *Phase*2*av*, *Phase*2*b*
⟨1⟩3.CASE *AcceptorReceiveAction*BY ⟨1⟩3, ⟨1⟩0a, ⟨1⟩0b, *SafeAcceptorIsAcceptor* DEF *AcceptorReceiveAction*, *F*
⟨1⟩4.CASE *AcceptorDisconnectAction*BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*
⟨1⟩5.CASE *LearnerAction*BY ⟨1⟩5 DEF *LearnerAction*, *LearnerDecide*, *LearnerRecv*
⟨1⟩6.CASE *FakeAcceptorAction*BY ⟨1⟩6 DEF *FakeAcceptorAction*, *FakeSend*, *Send*
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

LEMMA *VotesSentMonotone* $\triangleq$
$TypeOK \land Next \Rightarrow \forall A \in Acceptor : votesSent[A] \subseteq votesSent'[A]$
PROOF
⟨1⟩ SUFFICES ASSUME *TypeOK*, *Next*, NEW $A \in Acceptor$PROVE $votesSent[A] \subseteq votesSent'[A]$OBVIOUS
⟨1⟩0a. *TypeOK* OBVIOUS
⟨1⟩0b. *TypeOK*′ BY *TypeOKInvariant*
⟨1⟩1.CASE *ProposerAction*BY ⟨1⟩1 DEF *ProposerAction*, *Phase*1*a*, *Phase*1*c*, *Send*
⟨1⟩2.CASE *AcceptorSendAction*BY ⟨1⟩2, ⟨1⟩0a, ⟨1⟩0b DEF *AcceptorSendAction*, *Send*, *Phase*1*b*, *Phase*2*av*, *Ph*
⟨1⟩3.CASE *AcceptorReceiveAction*BY ⟨1⟩3, ⟨1⟩0a, ⟨1⟩0b DEF *AcceptorReceiveAction*, *Recv*, *TypeOK*
⟨1⟩4.CASE *AcceptorDisconnectAction*BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*
⟨1⟩5.CASE *LearnerAction*BY ⟨1⟩5 DEF *LearnerAction*, *LearnerDecide*, *LearnerRecv*
⟨1⟩6.CASE *FakeAcceptorAction*BY ⟨1⟩6 DEF *FakeAcceptorAction*, *FakeSend*, *Send*
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

LEMMA *VotesSentSpec*1*Invariant* $\triangleq$ $Next \land VotesSentSpec1 \Rightarrow VotesSentSpec1'$
PROOF
⟨1⟩ SUFFICES ASSUME
Next, *VotesSentSpec*1, NEW $A \in SafeAcceptor$, NEW $vote \in votesSent'[A]$
PROVE $VotedFor(vote.lr, A, vote.bal, vote.val)'$
BY DEF *VotesSentSpec*1
⟨1⟩ USE DEF *VotesSentSpec*1
⟨1⟩1.CASE *ProposerAction*BY ⟨1⟩1, *SafeAcceptorIsAcceptor* DEF *ProposerAction*, *Phase*1*a*, *Phase*1*c*, *Next*, *Se*
⟨1⟩2.CASE *AcceptorSendAction*
⟨2⟩.SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in SafeAcceptor$,
NEW $val \in Value$,
$\lor Phase1b(lrn, bal, acc)$
$\lor Phase2av(lrn, bal, acc, val)$

$\lor\ Phase2b(lrn,\ bal,\ acc,\ val)$
PROVE $VotedFor(vote.lr,\ A,\ vote.bal,\ vote.val)'$
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 $Send([type \mapsto$ "2b", $lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val])$,
$votesSent' = [votesSent$ EXCEPT $![acc] =$
$votesSent[acc] \cup \{[lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val]\}]$
PROVE $VotedFor(vote.lr,\ A,\ vote.bal,\ vote.val)'$
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 $vote \in votesSent[acc]$BY $\langle 3\rangle 2$, $\langle 4\rangle 2$, *MsgsMonotone*
$\langle 4\rangle 3$.CASE $vote \notin votesSent[acc]$
$\langle 5\rangle 1$. DEFINE $m0 \triangleq [type \mapsto$ "2b", $lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val]$
$\langle 5\rangle 2$. $m0 \in msgs'$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 $VotesSentSpec2Invariant \triangleq TypeOK \land Next \land VotesSentSpec2 \Rightarrow VotesSentSpec2'$
PROOF
$\langle 1\rangle$ SUFFICES ASSUME $TypeOK$, $Next$, $VotesSentSpec2$,
NEW $L \in Learner$, NEW $A \in SafeAcceptor$, NEW $B \in Ballot$, NEW $V \in Value$
PROVE $(VotedFor(L,\ A,\ B,\ V) \Rightarrow [lr \mapsto L,\ bal \mapsto B,\ val \mapsto V] \in votesSent[A])'$
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 $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in SafeAcceptor$,
NEW $val \in Value$,
$\lor\ Phase1b(lrn,\ bal,\ acc)$

$\lor\ Phase2av(lrn,\ bal,\ acc,\ val)$
$\lor\ Phase2b(lrn,\ bal,\ acc,\ val)$
PROVE $(VotedFor(L,\ A,\ B,\ V) \Rightarrow [lr \mapsto L,\ bal \mapsto B,\ val \mapsto V] \in votesSent[A])'$
BY $\langle 1\rangle 2$ DEF *AcceptorSendAction*
$\langle 2\rangle 1$.CASE $Phase1b(lrn,\ bal,\ acc)$BY $\langle 2\rangle 1$ DEF *Phase*1*b*
$\langle 2\rangle 2$.CASE $Phase2av(lrn,\ bal,\ acc,\ val)$BY $\langle 2\rangle 2$ DEF *Phase*2*av*
$\langle 2\rangle 3$.CASE $Phase2b(lrn,\ bal,\ acc,\ val)$
$\langle 3\rangle$ SUFFICES ASSUME $Send([type \mapsto$ "2b", $lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val])$,
$votesSent' = [votesSent$ EXCEPT $![acc] =$
$votesSent[acc] \cup \{[lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val]\}]$
PROVE $(VotedFor(L,\ A,\ B,\ V) \Rightarrow [lr \mapsto L,\ bal \mapsto B,\ val \mapsto V] \in votesSent[A])'$
BY $\langle 2\rangle 3$ DEF *Phase*2*b*
$\langle 3\rangle 1$. QED BY $\langle 1\rangle 0$b 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 $VotesSentSpec3Invariant \triangleq TypeOK \land Next \land VotesSentSpec3 \Rightarrow VotesSentSpec3'$
PROOF
$\langle 1\rangle$ SUFFICES ASSUME $TypeOK$, $Next$, $VotesSentSpec3$,
NEW $A \in SafeAcceptor$, NEW $B \in Ballot$,
NEW $V \in votesSent'[A]$,
$V.bal < B$
PROVE $(\exists\, P \in votesSent[A] : MaxVote(A,\ B,\ P) \land P.lr = V.lr \land V.bal \leq P.bal)'$
BY DEF *VotesSentSpec*3
$\langle 1\rangle$ USE DEF *VotesSentSpec*3
$\langle 1\rangle 0$a. *TypeOK*OBVIOUS
$\langle 1\rangle 0$b. *TypeOK*′BY *TypeOKInvariant*
$\langle 1\rangle 1$.CASE *ProposerAction*BY $\langle 1\rangle 1$ DEF *ProposerAction*, *Phase*1*a*, *Phase*1*c*, *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 $val \in Value$,
$\lor\ Phase1b(lrn,\ bal,\ acc)$
$\lor\ Phase2av(lrn,\ bal,\ acc,\ val)$
$\lor\ Phase2b(lrn,\ bal,\ acc,\ val)$
PROVE $(\exists\, P \in votesSent[A] : MaxVote(A,\ B,\ P) \land P.lr = V.lr \land V.bal \leq P.bal)'$
BY $\langle 1\rangle 2$ DEF *AcceptorSendAction*
$\langle 2\rangle 1$.CASE $Phase1b(lrn,\ bal,\ acc)$BY $\langle 2\rangle 1$ DEF *Phase*1*b*
$\langle 2\rangle 2$.CASE $Phase2av(lrn,\ bal,\ acc,\ val)$BY $\langle 2\rangle 2$ DEF *Phase*2*av*
$\langle 2\rangle 3$.CASE $Phase2b(lrn,\ bal,\ acc,\ val)$

```
    ⟨3⟩ SUFFICES ASSUME votesSent′ = [votesSent EXCEPT ![acc] =
                                        votesSent[acc] ∪ {[lr ↦ lrn, bal ↦ bal, val ↦ val]}]
                 PROVE (∃ P ∈ votesSent[A] : MaxVote(A, B, P) ∧ P.lr = V.lr ∧ V.bal ≤ P.bal)′
      BY ⟨2⟩3 DEF Phase2b
    ⟨3⟩1.CASE A = acc
      ⟨4⟩0. DEFINE v0 ≜ [lr ↦ lrn, bal ↦ bal, val ↦ val]
      ⟨4⟩1. v0 ∈ votesSent[A]′BY ⟨3⟩1, ⟨1⟩0b DEF TypeOK
      ⟨4⟩2.CASE V ∈ votesSent[A]BY ⟨4⟩2
      ⟨4⟩3.CASE V ∉ votesSent[A]
        ⟨5⟩0. V = v0BY ⟨4⟩3, ⟨3⟩1
        ⟨5⟩1.CASE ∀ P ∈ votesSent[A] : P.lr = lrn ⇒ P.bal ≥ B
          ⟨6⟩1. WITNESS v0 ∈ votesSent[A]′
          ⟨6⟩2. QED BY ⟨3⟩1, ⟨5⟩1, ⟨1⟩0b, ⟨5⟩0 DEF Ballot, TypeOK, MaxVote
        ⟨5⟩2.CASE ∃ P ∈ votesSent[A] : P.lr = lrn ∧ P.bal < B
          ⟨6⟩1. PICK P ∈ votesSent[A] : P.lr = lrn ∧ P.bal < BBY ⟨5⟩2
          ⟨6⟩2. PICK Pmax ∈ votesSent[A] : MaxVote(A, B, Pmax) ∧ Pmax.lr = lrn ∧ P.bal ≤ Pmax.balBY ⟨6
          ⟨6⟩3. Pmax ∈ votesSent[A]′BY ⟨3⟩1, ⟨6⟩2
          ⟨6⟩4.CASE Pmax.bal < bal
            ⟨7⟩1. WITNESS v0 ∈ votesSent[A]′
            ⟨7⟩2. SUFFICES MaxVote(A, B, v0)′BY ⟨5⟩0 DEF Ballot
            ⟨7⟩3. QED BY ⟨5⟩0, ⟨6⟩4, ⟨6⟩2, ⟨1⟩0b, ⟨3⟩1 DEF Ballot, TypeOK, MaxVote
          ⟨6⟩5.CASE bal ≤ Pmax.bal
            ⟨7⟩1. WITNESS Pmax ∈ votesSent[A]′
            ⟨7⟩20. QED BY ⟨5⟩0, ⟨6⟩5, ⟨6⟩2, ⟨1⟩0b, ⟨3⟩1 DEF Ballot, TypeOK
          ⟨6⟩20. QED BY ⟨6⟩4, ⟨6⟩5 DEF Ballot, TypeOK
        ⟨5⟩3. QED BY ⟨5⟩1, ⟨5⟩2 DEF Ballot, TypeOK
      ⟨4⟩4. QED BY ⟨4⟩2, ⟨4⟩3
    ⟨3⟩2.CASE A ≠ accBY ⟨3⟩2
    ⟨3⟩3. QED BY ⟨3⟩1, ⟨3⟩2
  ⟨2⟩5. QED BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3.CASE AcceptorReceiveActionBY ⟨1⟩3 DEF AcceptorReceiveAction, Recv, Next
⟨1⟩4.CASE AcceptorDisconnectActionBY ⟨1⟩4 DEF AcceptorDisconnectAction, Disconnect, Next
⟨1⟩5.CASE LearnerActionBY ⟨1⟩5 DEF LearnerAction, LearnerRecv, LearnerDecide, Next
⟨1⟩6.CASE FakeAcceptorActionBY ⟨1⟩6 DEF FakeAcceptorAction, FakeSend, Send
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF Next

LEMMA VotesSentSpec4Invariant ≜
    TypeOK ∧ Next ∧ MsgInv ∧ ReceivedSpec ∧
    VotesSentSpec1 ∧ 2avSentSpec2 ∧ 2avSentSpec3 ∧ VotesSentSpec4 ⇒
    VotesSentSpec4′
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, MsgInv, ReceivedSpec, VotesSentSpec1,
                    2avSentSpec2, 2avSentSpec3, VotesSentSpec4,
                    NEW A ∈ SafeAcceptor,
                    NEW vote1 ∈ votesSent′[A], NEW vote2 ∈ votesSent′[A],
```

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

$\land \ m1.bal = bal$
$\land \ m1.val = vote1.val$
BY $\langle 4\rangle 2$, $\langle 5\rangle 3$ DEF *VotesSentSpec*1, *VotedFor*
$\langle 5\rangle 5$. PICK $Q1 \in ByzQuorum$:
$\land \ [lr \mapsto vote1.lr,\ q \mapsto Q1] \in TrustLive$
$\land \ \forall\, ba \in Q1$:
$\exists\, m2av \in received[acc]$:
$\land\, m2av.type =$ "2av"
$\land\, m2av.lr = vote1.lr$
$\land\, m2av.acc = ba$
$\land\, m2av.bal = bal$
$\land\, m2av.val = vote1.val$
$\langle 6\rangle 1$. $\exists\, Q1 \in ByzQuorum$:
$\land\, [lr \mapsto m1.lr,\ q \mapsto Q1] \in TrustLive$
$\land\, \forall\, ba \in Q1$:
$\exists\, m2av \in received[m1.acc]$:
$\land\, m2av.type =$ "2av"
$\land\, m2av.lr = m1.lr$
$\land\, m2av.acc = ba$
$\land\, m2av.bal = m1.bal$
$\land\, m2av.val = m1.val$
BY $\langle 5\rangle 4$, $\langle 3\rangle 1$ DEF *MsgInv*2*b*, *TypeOK*
$\langle 6\rangle 2$. QED BY $\langle 5\rangle 4$, $\langle 6\rangle 1$, $\langle 3\rangle 1$
$\langle 5\rangle 6$. $\langle vote1.lr,\ lrn\rangle \in Ent$BY $\langle 5\rangle 3$
$\langle 5\rangle 7$. PICK $S \in SafeAcceptor : S \in Q1 \land S \in Q2$BY $\langle 5\rangle 2$, $\langle 5\rangle 5$, $\langle 5\rangle 6$, $\langle 5\rangle 0$, *EntanglementTrustLive*
$\langle 5\rangle 8$. PICK $m2av1 \in received[acc]$:
$\land\, m2av1.type =$ "2av"
$\land\, m2av1.lr = vote1.lr$
$\land\, m2av1.acc = S$
$\land\, m2av1.bal = bal$
$\land\, m2av1.val = vote1.val$
BY $\langle 5\rangle 7$, $\langle 5\rangle 5$
$\langle 5\rangle 9$. $\land\, m2av1 \in msgs$
$\land\, m2av1.type =$ "2av"
$\land\, m2av1.lr = vote1.lr$
$\land\, m2av1.acc = S$
$\land\, m2av1.bal = bal$
$\land\, m2av1.val = vote1.val$
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 2*avSentSpec*2, *Proposed*
$\langle 5\rangle 11$. PICK $m2av2 \in received[acc]$:
$\land\, m2av2.type =$ "2av"
$\land\, m2av2.lr = lrn$
$\land\, m2av2.acc = S$

$\land\ m2av2.bal = bal$
$\land\ m2av2.val = val$
BY $\langle 5\rangle 7$, $\langle 5\rangle 2$

$\langle 5\rangle 12.\ \land\ m2av2 \in msgs$
$\land\ m2av2.type = \text{“2av”}$
$\land\ m2av2.lr = lrn$
$\land\ m2av2.acc = S$
$\land\ m2av2.bal = bal$
$\land\ m2av2.val = val$
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*

$\langle 5\rangle 14.\ vote1.val = val$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 $\langle 5\rangle 1$, $\langle 5\rangle 14$

$\langle 4\rangle 3.$CASE $vote1 \notin votesSent[A] \land vote2 \in votesSent[A]$

$\langle 5\rangle 0.\ vote2.lr \in Learner \land vote2.val \in Value$BY $\langle 4\rangle 3$ DEF *TypeOK*

$\langle 5\rangle 1.\ vote1 = [lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val]$BY $\langle 4\rangle 3$

$\langle 5\rangle 2.$ PICK $Q1 \in ByzQuorum$:
$\land\ [lr \mapsto lrn,\ q \mapsto Q1] \in TrustLive$
$\land\ \forall\, aa \in Q1 :$
$\exists\, m \in \{mm \in received[acc] :$
$\land\ mm.type = \text{“2av”}$
$\land\ mm.lr = lrn$
$\land\ mm.bal = bal\} :$
$\land\ m.val = val$
$\land\ m.acc = aa$
BY $\langle 5\rangle 1$, $\langle 2\rangle 3$ DEF *Phase2b*

$\langle 5\rangle 3.\ \langle lrn,\ vote2.lr\rangle \in Ent \land vote2.bal = bal$BY $\langle 5\rangle 1$

$\langle 5\rangle 4.$ PICK $m2 \in msgs$:
$\land\ m2.type = \text{“2b”}$
$\land\ m2.lr = vote2.lr$
$\land\ m2.acc = A$
$\land\ m2.bal = bal$
$\land\ m2.val = vote2.val$
BY $\langle 4\rangle 3$, $\langle 5\rangle 3$ DEF *VotesSentSpec*1, *VotedFor*

$\langle 5\rangle 5.$ PICK $Q2 \in ByzQuorum$:
$\land\ [lr \mapsto vote2.lr,\ q \mapsto Q2] \in TrustLive$
$\land\ \forall\, ba \in Q2 :$
$\exists\, m2av \in received[acc] :$
$\land\ m2av.type = \text{“2av”}$
$\land\ m2av.lr = vote2.lr$
$\land\ m2av.acc = ba$
$\land\ m2av.bal = bal$
$\land\ m2av.val = vote2.val$

$\langle 6\rangle 1.\ \exists\, Q2 \in ByzQuorum$:

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

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

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

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

LEMMA $2avSentSpec2Invariant \triangleq Next \land 2avSentSpec2 \Rightarrow 2avSentSpec2'$
PROOF
⟨1⟩ 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 $2avSentSpec2$
⟨1⟩ USE DEF $2avSentSpec2$
⟨1⟩1.CASE $ProposerAction$BY ⟨1⟩1 DEF $ProposerAction$, $Phase1a$, $Phase1c$, $Next$, $Send$
⟨1⟩2.CASE $AcceptorSendAction$
⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in SafeAcceptor$,
NEW $val \in Value$,
$\lor\ Phase1b(lrn,\ bal,\ acc)$
$\lor\ Phase2av(lrn,\ bal,\ acc,\ val)$
$\lor\ Phase2b(lrn,\ bal,\ acc,\ val)$
PROVE $([lr \mapsto L,\ bal \mapsto B,\ val \mapsto V] \in 2avSent[A])'$
BY ⟨1⟩2 DEF $AcceptorSendAction$
⟨2⟩1.CASE $Phase1b(lrn,\ bal,\ acc)$BY ⟨2⟩1 DEF $Phase1b$
⟨2⟩2.CASE $Phase2av(lrn,\ bal,\ acc,\ val)$
⟨3⟩ SUFFICES ASSUME $Send([type \mapsto$ "2av", $lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val])$,
$2avSent' = [2avSent$ 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 ⟨2⟩2 DEF $Phase2av$
⟨3⟩1. QED OBVIOUS
⟨2⟩3.CASE $Phase2b(lrn,\ bal,\ acc,\ val)$BY ⟨2⟩3 DEF $Phase2b$
⟨2⟩5. QED BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3.CASE $AcceptorReceiveAction$BY ⟨1⟩3 DEF $AcceptorReceiveAction$, $Recv$, $Next$
⟨1⟩4.CASE $AcceptorDisconnectAction$BY ⟨1⟩4 DEF $AcceptorDisconnectAction$, $Disconnect$, $Next$
⟨1⟩5.CASE $LearnerAction$BY ⟨1⟩5 DEF $LearnerAction$, $LearnerRecv$, $LearnerDecide$, $Next$
⟨1⟩6.CASE $FakeAcceptorAction$BY ⟨1⟩6 DEF $FakeAcceptorAction$, $FakeSend$, $Send$

⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

LEMMA $2avSentSpec3Invariant \triangleq Next \land ConnectedSpec \land 2avSentSpec3 \Rightarrow 2avSentSpec3'$
PROOF
⟨1⟩ 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 $2avSentSpec3$
⟨1⟩ USE DEF $2avSentSpec3$
⟨1⟩1.CASE *ProposerAction*BY ⟨1⟩1 DEF *ProposerAction*, *Phase*1*a*, *Phase*1*c*, *Next*, *Send*
⟨1⟩2.CASE *AcceptorSendAction*
⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in SafeAcceptor$,
NEW $val \in Value$,
$\lor Phase1b(lrn, bal, acc)$
$\lor Phase2av(lrn, bal, acc, val)$
$\lor Phase2b(lrn, bal, acc, val)$
PROVE $V1 = V2$
BY ⟨1⟩2 DEF *AcceptorSendAction*
⟨2⟩1.CASE $Phase1b(lrn, bal, acc)$BY ⟨2⟩1 DEF *Phase*1*b*
⟨2⟩2.CASE $Phase2av(lrn, bal, acc, val)$
⟨3⟩ SUFFICES
ASSUME NEW $v \in Value$,
$\forall P \in \{p \in 2avSent[acc] : p.bal = bal \land \langle p.lr, lrn\rangle \in connected[acc]\} : P.val = v$,
$2avSent' = [2avSent$ EXCEPT $![acc] =$
$2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto v]\}]$
PROVE $V1 = V2$
BY ⟨2⟩2 DEF *Phase*2*av*
⟨3⟩1.CASE $A = acc$
⟨4⟩1.CASE $\land [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in 2avSent[A]$
$\land [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent[A]$
BY ⟨4⟩1, ⟨3⟩1
⟨4⟩3.CASE $\land [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \notin 2avSent[A]$
$\land [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent[A]$
⟨5⟩1. $\langle L2, L1\rangle \in Ent$BY *EntanglementSym*
⟨5⟩2. QED BY ⟨4⟩3, ⟨3⟩1, ⟨5⟩1 DEF *ConnectedSpec*
⟨4⟩2.CASE $\land [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in 2avSent[A]$
$\land [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \notin 2avSent[A]$
BY ⟨4⟩2, ⟨3⟩1 DEF *ConnectedSpec*
⟨4⟩4.CASE $\land [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \notin 2avSent[A]$

$\wedge\, [lr \mapsto L2,\ bal \mapsto B,\ val \mapsto V2] \notin 2avSent[A]$
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 $LearnerDecide(lrn,\ bal)$
$\langle 3\rangle 0$a. $TypeOK$OBVIOUS
$\langle 3\rangle 0$b. $TypeOK'$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 $\langle 3\rangle 2$, $\langle 2\rangle 2$, $\langle 3\rangle 0$a, $\langle 3\rangle 0$b 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

⟨1⟩ 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*
⟨1⟩ USE DEF *ConnectedSpec*
⟨1⟩1.CASE *ProposerAction*BY ⟨1⟩1 DEF *ProposerAction*, *Phase*1*a*, *Phase*1*c*, *Next*
⟨1⟩2.CASE *AcceptorSendAction*BY ⟨1⟩2 DEF *AcceptorSendAction*, *Phase*1*b*, *Phase*2*b*, *Phase*2*av*, *Next*
⟨1⟩3.CASE *AcceptorReceiveAction*BY ⟨1⟩3 DEF *AcceptorReceiveAction*, *Recv*, *Next*
⟨1⟩4.CASE *AcceptorDisconnectAction*BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*, *Next*
⟨1⟩5.CASE *LearnerAction*BY ⟨1⟩5 DEF *LearnerAction*, *LearnerRecv*, *LearnerDecide*, *Next*
⟨1⟩6.CASE *FakeAcceptorAction*BY ⟨1⟩6 DEF *FakeAcceptorAction*, *FakeSend*, *Send*
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

LEMMA *MsgInvInvariant* $\triangleq$
$TypeOK \land MsgInv \land VotesSentSpec1 \land VotesSentSpec2 \land VotesSentSpec3 \land 2avSentSpec1 \land Next \Rightarrow MsgInv'$
PROOF
⟨1⟩ USE DEF *MsgInv*
⟨1⟩1b. ASSUME *TypeOK*, *VotesSentSpec*1, *VotesSentSpec*2, *VotesSentSpec*3, 2*avSentSpec*1, *Next*,
$\forall\, m \in msgs : m.acc \in SafeAcceptor \land m.type = \text{“1b”} \Rightarrow MsgInv1b(m)$,
NEW $m \in msgs'$, $m.acc \in SafeAcceptor$, $m.type = \text{“1b”}$
PROVE $MsgInv1b(m)'$
⟨2⟩0. *TypeOK*BY ⟨1⟩1b
⟨2⟩0a. *TypeOK*′BY ⟨1⟩1b, *TypeOKInvariant*
⟨2⟩0b. $m \in Message$BY ⟨2⟩0a DEF *TypeOK*
⟨2⟩0c. $maxBal \in [Learner \times Acceptor \to Ballot]$BY ⟨1⟩1b DEF *TypeOK*
⟨2⟩0d. $maxBal' \in [Learner \times Acceptor \to Ballot]$BY ⟨2⟩0a DEF *TypeOK*
⟨2⟩0e. $m.type = \text{“1b”}$BY ⟨1⟩1b
⟨2⟩0f. $m.bal \in Ballot$BY ⟨2⟩0b, ⟨2⟩0e DEF *Message*, *Ballot*
⟨2⟩0g. $maxBal[m.lr, m.acc] \in Ballot$BY ⟨2⟩0b, ⟨2⟩0c, ⟨2⟩0e DEF *Message*
⟨2⟩0h. $maxBal'[m.lr, m.acc] \in Ballot$BY ⟨2⟩0b, ⟨2⟩0d, ⟨2⟩0e DEF *Message*
⟨2⟩0i. $maxBal[m.lr, m.acc] \leq maxBal'[m.lr, m.acc]$BY ⟨1⟩1b, ⟨2⟩0b, *MaxBalMonotone* DEF *TypeOK*, *Messa*
⟨2⟩1.CASE *ProposerAction*
⟨3⟩ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $proposer \in Ballot$, NEW $val \in Value$,
$\lor\ Phase1a(lrn, proposer)$
$\lor\ Phase1c(lrn, proposer, val)$
PROVE $MsgInv1b(m)'$
BY ⟨2⟩1, *ValueNotEmpty* DEF *ProposerAction*
⟨3⟩1.CASE *Phase*1*a*$(lrn, proposer)$
⟨4⟩1. $m \in msgs$BY ⟨3⟩1, ⟨2⟩0e DEF *Phase*1*a*, *Send*
⟨4⟩2. QED BY ⟨1⟩1b, ⟨4⟩1, ⟨3⟩1 DEF *Phase*1*a*, *MsgInv*1*b*
⟨3⟩2.CASE *Phase*1*c*$(lrn, proposer, val)$
⟨4⟩1. $m \in msgs$BY ⟨3⟩2, ⟨2⟩0e DEF *Phase*1*c*, *Send*, *TypeOK*

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

⟨4⟩10. QED BY ⟨4⟩1, ⟨4⟩5, ⟨4⟩6 DEF *MsgInv*1*b*
⟨3⟩2.CASE *Phase*2*av*(*lrn*, *bal*, *acc*, *val*)
⟨4⟩ SUFFICES
ASSUME *maxBal*[*lrn*, *acc*] ≤ *bal*,
Send([*type* ↦ "2av", *lr* ↦ *lrn*, *acc* ↦ *acc*, *bal* ↦ *bal*, *val* ↦ *val*]),
$2avSent' = [2avSent \text{ EXCEPT } ![acc] = 2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
PROVE *MsgInv*1*b*(*m*)′
BY ⟨3⟩2 DEF *Phase*2*av*
⟨4⟩1. *m* ∈ *msgs*BY ⟨2⟩0e DEF *Send*
⟨4⟩1a. *m.acc* ∈ *Acceptor*BY ⟨4⟩1, *MessageType*, ⟨2⟩0e, ⟨2⟩0 DEF *TypeOK*
⟨4⟩2. (*m.bal* ≤ *maxBal*[*m.lr*, *m.acc*])′BY ⟨1⟩1b, ⟨4⟩1, ⟨3⟩2 DEF *Phase*2*av*, *Send*, *MsgInv*1*b*
⟨4⟩4. (*m.votes* = {*p* ∈ *votesSent*[*m.acc*] : *MaxVote*(*m.acc*, *m.bal*, *p*)})′
BY ⟨1⟩1b, ⟨4⟩1, ⟨3⟩2 DEF *Phase*2*av*, *Send*, *MsgInv*1*b*
⟨4⟩5. (*m.proposals* = {*p* ∈ 2*avSent*[*m.acc*] : *p.bal* < *m.bal* ∧ *p.lr* = *m.lr*})′
⟨5⟩1.CASE *m.acc* ≠ *acc*
⟨6⟩1. 2*avSent*′[*m.acc*] = 2*avSent*[*m.acc*]BY ⟨3⟩2, ⟨4⟩1, ⟨5⟩1, ⟨2⟩0, ⟨2⟩0e, *MessageType* DEF *Phase*2*b*,
⟨6⟩2. QED BY ⟨6⟩1, ⟨4⟩1, ⟨1⟩1b, ⟨2⟩0e DEF *MsgInv*1*b*
⟨5⟩2.CASE *m.acc* = *acc*
⟨6⟩1. *m.bal* ≤ *maxBal*[*m.lr*, *m.acc*]BY ⟨1⟩1b, ⟨4⟩1 DEF *MsgInv*1*b*
⟨6⟩3. *m.bal* ∈ *Ballot*BY ⟨2⟩0a, *MessageType* DEF *TypeOK*
⟨6⟩5. SUFFICES {*p* ∈ 2*avSent*[*acc*] : *p.bal* < *m.bal* ∧ *p.lr* = *m.lr*} = {*p* ∈ 2*avSent*′[*acc*] : *p.bal* < *m.bal* ∧ *p.lr* = *m.lr*}
BY ⟨4⟩1, ⟨2⟩0e, ⟨1⟩1b, ⟨5⟩2 DEF *MsgInv*1*b*
⟨6⟩6. SUFFICES ASSUME NEW *p* ∈ 2*avSent*′[*acc*], *p.bal* < *m.bal*, *p.lr* = *m.lr*
PROVE *p* ∈ 2*avSent*[*acc*]BY ⟨2⟩0a, *SafeAcceptorIsAcceptor* DEF *TypeOK*
⟨6⟩7.CASE *p* ∈ 2*avSent*[*acc*]BY ⟨6⟩7
⟨6⟩8.CASE *p* ∉ 2*avSent*[*acc*]
⟨7⟩1. *p* = [*lr* ↦ *lrn*, *bal* ↦ *bal*, *val* ↦ *val*]BY ⟨6⟩8, ⟨2⟩0a, *SafeAcceptorIsAcceptor* DEF *TypeOK*
⟨7⟩2. *maxBal*[*m.lr*, *m.acc*] ≤ *bal*BY ⟨5⟩2, ⟨7⟩1, ⟨6⟩6
⟨7⟩4. *m.bal* ≤ *bal*BY ⟨6⟩1, ⟨7⟩2, ⟨6⟩3, ⟨2⟩0g, *BallotLeqTrans*
⟨7⟩10. QED BY ⟨7⟩1, ⟨7⟩4, ⟨6⟩6, ⟨6⟩3, *BallotLeNotLeq*
⟨6⟩10. QED BY ⟨6⟩7, ⟨6⟩8
⟨5⟩3. QED BY ⟨5⟩1, ⟨5⟩2
⟨4⟩10. QED BY ⟨4⟩2, ⟨4⟩4, ⟨4⟩5 DEF *MsgInv*1*b*
⟨3⟩3.CASE *Phase*2*b*(*lrn*, *bal*, *acc*, *val*)
⟨4⟩ SUFFICES
ASSUME ∀ *L* ∈ *Learner* : *maxBal*[*L*, *acc*] ≤ *bal*,
Send([*type* ↦ "2b", *lr* ↦ *lrn*, *acc* ↦ *acc*, *bal* ↦ *bal*, *val* ↦ *val*]),
$votesSent' = [votesSent \text{ EXCEPT } ![acc] = votesSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
PROVE *MsgInv*1*b*(*m*)′
BY ⟨3⟩3 DEF *Phase*2*b*
⟨4⟩1. *m* ∈ *msgs*BY ⟨2⟩0e DEF *Send*
⟨4⟩1a. *m.acc* ∈ *Acceptor*BY ⟨4⟩1, *MessageType*, ⟨2⟩0e, ⟨2⟩0 DEF *TypeOK*

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

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

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

$\land \forall\, ba \in Q :$
$\exists\, m1b \in received[m.acc] :$
$\land\ m1b.type = \text{“1b”}$
$\land\ m1b.lr = m.lr$
$\land\ m1b.acc = ba$
$\land\ m1b.bal = m.bal)'$

$\langle 7\rangle 1.$ PICK $Q0 \in ByzQuorum :$
$\land\ [lr \mapsto m.lr,\ q \mapsto Q0] \in TrustLive$
$\land\ \forall\, ba \in Q0 :$
$\exists\, m1b \in received[m.acc] :$
$\land\ m1b.type = \text{“1b”}$
$\land\ m1b.lr = m.lr$
$\land\ m1b.acc = ba$
$\land\ m1b.bal = m.bal$
BY $\langle 1\rangle$2av, $\langle 3\rangle 1$, $\langle 2\rangle$0e DEF $MsgInv2av$

$\langle 7\rangle 2.$ WITNESS $Q0 \in ByzQuorum$

$\langle 7\rangle 3.$ QED BY $\langle 1\rangle$2av, $\langle 7\rangle 1$, $ReceivedMonotone$, $MessageType$, $\langle 3\rangle 1$ DEF $MsgInv2av$, $TypeOK$

$\langle 3\rangle 20.$ QED BY $\langle 1\rangle$2av, $\langle 2\rangle 4$, $\langle 3\rangle 6$, $MessageType$, $ReceivedMonotone$ DEF $MsgInv2av$, $AcceptorReceiveActio$

$\langle 2\rangle 5.$CASE $AcceptorDisconnectAction$BY $\langle 1\rangle$2av, $\langle 2\rangle 5$ DEF $AcceptorDisconnectAction$, $Disconnect$, $MsgInv2a$

$\langle 2\rangle 6.$CASE $LearnerAction$BY $\langle 1\rangle$2av, $\langle 2\rangle 6$ DEF $LearnerAction$, $LearnerRecv$, $LearnerDecide$, $MsgInv2av$, Ne

$\langle 2\rangle 7.$CASE $FakeAcceptorAction$
BY $\langle 1\rangle$2av, $\langle 2\rangle 7$, $SafeAcceptorAssumption$ DEF $FakeAcceptorAction$, $FakeSend$, $MsgInv2av$, $Send$

$\langle 2\rangle 8.$ QED BY $\langle 1\rangle$2av, $\langle 2\rangle$0b, $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 4$, $\langle 2\rangle 5$, $\langle 2\rangle 6$, $\langle 2\rangle 7$ DEF $Next$

$\langle 1\rangle$2b. ASSUME $TypeOK$, $Next$, $\forall\, m \in msgs : m.acc \in SafeAcceptor \land m.type = \text{“2b”} \Rightarrow MsgInv2b(m)$,
NEW $m \in msgs'$, $m.acc \in SafeAcceptor$, $m.type = \text{“2b”}$
PROVE $MsgInv2b(m)'$

$\langle 2\rangle$0a. $TypeOK$BY $\langle 1\rangle$2b

$\langle 2\rangle$0b. $TypeOK'$BY $\langle 1\rangle$2b, $TypeOKInvariant$

$\langle 2\rangle$0c. $m \in Message$BY $\langle 2\rangle$0b DEF $TypeOK$

$\langle 2\rangle$0d. $m.acc \in SafeAcceptor$BY $\langle 1\rangle$2b

$\langle 2\rangle$0e. $m.type = \text{“2b”}$BY $\langle 1\rangle$2b

$\langle 2\rangle 1.$CASE $ProposerAction$

$\langle 3\rangle 1.$ $m \in msgs$BY $\langle 2\rangle 1$, $\langle 2\rangle$0e DEF $ProposerAction$, $Phase1a$, $Phase1c$, $Send$

$\langle 3\rangle 10.$ 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, $\langle 3\rangle 1$
DEF $TypeOK$, $ProposerAction$, $Phase1a$, $Phase1c$, $MsgInv2b$, $Next$, $Send$

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

$\langle 3\rangle$ HIDE DEF $Next$

$\langle 3\rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in SafeAcceptor$,
NEW $val \in Value$,
$\lor\ Phase1b(lrn,\ bal,\ acc)$
$\lor\ Phase2av(lrn,\ bal,\ acc,\ val)$
$\lor\ Phase2b(lrn,\ bal,\ acc,\ val)$
PROVE $MsgInv2b(m)'$

BY ⟨2⟩2 DEF *AcceptorSendAction*
⟨3⟩1.CASE *Phase*1*b*(*lrn*, *bal*, *acc*)
⟨4⟩1. $m \in msgs$BY ⟨3⟩1, ⟨2⟩0a, ⟨2⟩0e DEF *Phase*1*b*, *Send*, *TypeOK*
⟨4⟩2. QED BY ⟨1⟩2b, ⟨3⟩1, ⟨2⟩0a, ⟨2⟩0b, ⟨2⟩0e, ⟨4⟩1 DEF *Phase*1*b*, *MsgInv*2*b*, *Send*, *TypeOK*
⟨3⟩2.CASE *Phase*2*av*(*lrn*, *bal*, *acc*, *val*)
⟨4⟩1. $m \in msgs$BY ⟨3⟩2, ⟨2⟩0a, ⟨2⟩0e DEF *Phase*2*av*, *Send*, *TypeOK*
⟨4⟩2. QED BY ⟨1⟩2b, ⟨3⟩2, ⟨2⟩0a, ⟨2⟩0d, ⟨2⟩0e, ⟨4⟩1 DEF *Phase*2*av*, *MsgInv*2*b*, *Send*, *TypeOK*
⟨3⟩3.CASE *Phase*2*b*(*lrn*, *bal*, *acc*, *val*)
⟨4⟩1.CASE $m \in msgs$
⟨5⟩1. $([lr \mapsto m.lr,\ bal \mapsto m.bal,\ val \mapsto m.val] \in votesSent[m.acc])'$
BY ⟨1⟩2b, ⟨2⟩0e, ⟨4⟩1, *MessageType*, *VotesSentMonotone* DEF *MsgInv*2*b*, *TypeOK*
⟨5⟩2. $(\exists\, Q \in ByzQuorum :$
$\land [lr \mapsto m.lr,\ q \mapsto Q] \in TrustLive$
$\land \forall\, ba \in Q :$
$\exists\, m2av \in received[m.acc] :$
$\land\ m2av.type = \text{“2av”}$
$\land\ m2av.lr = m.lr$
$\land\ m2av.acc = ba$
$\land\ m2av.bal = m.bal$
$\land\ m2av.val = m.val)'$
BY ⟨1⟩2b, ⟨3⟩3, ⟨2⟩0a, ⟨2⟩0b, ⟨2⟩0d, ⟨2⟩0e, ⟨4⟩1 DEF *Phase*2*b*, *MsgInv*2*b*, *Send*, *TypeOK*
⟨5⟩3. QED BY ⟨5⟩1, ⟨5⟩2 DEF *MsgInv*2*b*
⟨4⟩2.CASE $m \notin msgs$
⟨5⟩ SUFFICES
ASSUME NEW $Q \in ByzQuorum$,
$[lr \mapsto lrn,\ q \mapsto Q] \in TrustLive$,
$\forall\, aa \in Q :$
$\exists\, m\_1 \in \{mm \in received[acc] :$
$\land\ mm.type = \text{“2av”}$
$\land\ mm.lr = lrn$
$\land\ mm.bal = bal\} :$
$\land\ m\_1.val = val$
$\land\ m\_1.acc = aa,$
$Send([type \mapsto \text{“2b”},\ lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val]),$
$votesSent' = [votesSent \text{ EXCEPT } ![acc] =$
$votesSent[acc] \cup \{[lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val]\}]$
PROVE $MsgInv2b(m)'$
BY ⟨3⟩3 DEF *Phase*2*b*
⟨5⟩1. $m = [type \mapsto \text{“2b”},\ lr \mapsto lrn,\ acc \mapsto acc,\ bal \mapsto bal,\ val \mapsto val]$BY ⟨4⟩2 DEF *Send*
⟨5⟩1e. UNCHANGED *received*BY ⟨3⟩3 DEF *Phase*2*b*
⟨5⟩2. $([lr \mapsto m.lr,\ bal \mapsto m.bal,\ val \mapsto m.val] \in votesSent[m.acc])'$
BY ⟨5⟩1, ⟨2⟩0a, ⟨2⟩0b, ⟨2⟩0e, *MessageType* DEF *TypeOK*
⟨5⟩3. $(\exists\, Q\_1 \in ByzQuorum :$
$\land [lr \mapsto m.lr,\ q \mapsto Q\_1] \in TrustLive$
$\land \forall\, ba \in Q\_1 :$

$\exists\, m2av \in received[m.acc] :$
$\land\, m2av.type = \text{“2av”}$
$\land\, m2av.lr = m.lr$
$\land\, m2av.acc = ba$
$\land\, m2av.bal = m.bal$
$\land\, m2av.val = m.val)'$

$\langle 6\rangle 1$. WITNESS $Q \in ByzQuorum$

$\langle 6\rangle 2$. QED BY $\langle 5\rangle 1$, $\langle 5\rangle 1$e, $\langle 2\rangle 0$a 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 $m0 \in msgs$,
$received' = [received$ EXCEPT $![acc] = received[acc] \cup \{m0\}]$,
UNCHANGED $\langle msgs, maxBal, 2avSent, votesSent, connected,$
$receivedByLearner, decision\rangle$
PROVE $MsgInv2b(m)'$

BY $\langle 2\rangle 4$, $\langle 2\rangle 0$b DEF $AcceptorReceiveAction$, $Recv$, $TypeOK$

$\langle 3\rangle 2$. $m \in msgs$BY $\langle 3\rangle 0$, $\langle 1\rangle 2$b

$\langle 3\rangle 2$a. $m \in Message$BY $\langle 3\rangle 2$, $\langle 1\rangle 2$b DEF $TypeOK$

$\langle 3\rangle 2$b. $TypeOK$BY $\langle 1\rangle 2$b DEF $Phase2b$

$\langle 3\rangle 2$c. $TypeOK'$BY $\langle 1\rangle 2$b, $\langle 3\rangle 2$b, $TypeOKInvariant$

$\langle 3\rangle 3$. $[lr \mapsto m.lr,\ bal \mapsto m.bal,\ val \mapsto m.val] \in votesSent'[m.acc]$BY $\langle 3\rangle 0$, $\langle 1\rangle 2$b DEF $MsgInv2b$

$\langle 3\rangle 5$. PICK $Q0 \in ByzQuorum :$
$\land\, [lr \mapsto m.lr,\ q \mapsto Q0] \in TrustLive$
$\land\, \forall\, ba \in Q0 :$
$\exists\, m2av \in received[m.acc] :$
$\land\, m2av.type = \text{“2av”}$
$\land\, m2av.lr = m.lr$
$\land\, m2av.acc = ba$
$\land\, m2av.bal = m.bal$
$\land\, m2av.val = m.val$BY $\langle 1\rangle 2$b, $\langle 2\rangle 0$e, $\langle 3\rangle 2$ DEF $MsgInv2b$

$\langle 3\rangle 7$. $(\exists\, Q \in ByzQuorum :$
$\land\, [lr \mapsto m.lr,\ q \mapsto Q] \in TrustLive$
$\land\, \forall\, ba \in Q :$
$\exists\, m2av \in received[m.acc] :$
$\land\, m2av.type = \text{“2av”}$
$\land\, m2av.lr = m.lr$
$\land\, m2av.acc = ba$
$\land\, m2av.bal = m.bal$
$\land\, m2av.val = m.val)'$

$\langle 4\rangle 0$. WITNESS $Q0 \in ByzQuorum$

$\langle 4\rangle 1$. QED BY $\langle 1\rangle 2$b, $\langle 3\rangle 5$, $\langle 3\rangle 2$b, $\langle 3\rangle 2$c, $MessageType$, $ReceivedMonotone$ DEF $TypeOK$

⟨3⟩8. QED BY ⟨3⟩3, ⟨3⟩7 DEF *MsgInv*2*b*
⟨2⟩5.CASE *AcceptorDisconnectAction*BY ⟨1⟩2b, ⟨2⟩5 DEF *AcceptorDisconnectAction*, *Disconnect*, *MsgInv*2*b*,
⟨2⟩6.CASE *LearnerAction*BY ⟨1⟩2b, ⟨2⟩6 DEF *LearnerAction*, *LearnerRecv*, *LearnerDecide*, *MsgInv*2*b*, *Next*
⟨2⟩7.CASE *FakeAcceptorAction*
BY ⟨1⟩2b, ⟨2⟩7, *SafeAcceptorAssumption* DEF *FakeAcceptorAction*, *FakeSend*, *MsgInv*2*b*, *Send*
⟨2⟩8. QED BY ⟨1⟩2b, ⟨2⟩0a, ⟨2⟩1, ⟨2⟩2, ⟨2⟩4, ⟨2⟩5, ⟨2⟩6, ⟨2⟩7 DEF *Next*
⟨1⟩3. QED BY ⟨1⟩1b, ⟨1⟩2av, ⟨1⟩2b

$$
\begin{array}{l}
\mathit{CannotDecide}(Q,\ L,\ B,\ V) \triangleq \\
\quad \exists\, A \in \mathit{SafeAcceptor} : \\
\qquad \land\ A \in Q \\
\qquad \land\ \exists\, L0 \in \mathit{Learner} : \mathit{LeftBallot}(L0,\ A,\ B) \quad \textit{TODO}\text{: check if used} \\
\qquad \land\ \neg\, \mathit{VotedFor}(L,\ A,\ B,\ V)
\end{array}
$$

$$
\begin{array}{l}
\mathit{HeterogeneousSpec} \triangleq \\
\quad \forall\, L1,\ L2 \in \mathit{Learner} : \\
\quad \forall\, B1,\ B2 \in \mathit{Ballot} : \\
\quad \forall\, V1,\ V2 \in \mathit{Value} : \\
\quad \forall\, A2 \in \mathit{SafeAcceptor} : \\
\quad \forall\, Q \in \mathit{ByzQuorum} : \\
\quad \forall\, M \in \mathit{msgs} : \\
\qquad \land\ \langle L1,\ L2\rangle \in \mathit{Ent} \\
\qquad \land\ [\mathit{lr} \mapsto L1,\ q \mapsto Q] \in \mathit{TrustLive} \\
\qquad \land\ M.\mathit{type} = \text{“}\mathsf{2av}\text{”} \land M.\mathit{lr} = L2 \land M.\mathit{acc} = A2 \land M.\mathit{bal} = B2 \land M.\mathit{val} = V2 \\
\qquad \land\ B1 < B2 \\
\qquad \land\ V1 \neq V2 \\
\qquad \Rightarrow \\
\qquad \mathit{CannotDecide}(Q,\ L1,\ B1,\ V1)
\end{array}
$$

$$
\begin{array}{l}
\text{LEMMA } \mathit{HeterogeneousSpecInvariant} \triangleq \\
\quad \mathit{TypeOK} \land \mathit{Next} \land \mathit{ReceivedSpec} \land \\
\quad 2\mathit{avSentSpec}1 \land \\
\quad \mathit{VotesSentSpec}2 \land \mathit{VotesSentSpec}3 \land \mathit{VotesSentSpec}4 \land \\
\quad \mathit{ConnectedSpec} \land \mathit{MsgInv} \land \\
\quad \mathit{HeterogeneousSpec} \Rightarrow \mathit{HeterogeneousSpec}'
\end{array}
$$

PROOF

⟨1⟩ SUFFICES ASSUME *TypeOK*, *Next*, *ReceivedSpec*, 2*avSentSpec*1, *VotesSentSpec*2, *VotesSentSpec*3, *VotesSer*
ConnectedSpec, *MsgInv*, *HeterogeneousSpec*,
NEW $L1 \in \mathit{Learner}$, NEW $L2 \in \mathit{Learner}$,
NEW $B1 \in \mathit{Ballot}$, NEW $B2 \in \mathit{Ballot}$,
NEW $V1 \in \mathit{Value}$, NEW $V2 \in \mathit{Value}$,
NEW $A2 \in \mathit{SafeAcceptor}$,
NEW $Q1 \in \mathit{ByzQuorum}$,
NEW $m \in \mathit{msgs}'$,
$\langle L1,\ L2\rangle \in \mathit{Ent}$,
$[\mathit{lr} \mapsto L1,\ q \mapsto Q1] \in \mathit{TrustLive}$,

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

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

$\land\ p.bal \leq c$
$\land\ (p.bal = c) \Rightarrow (p.val = V2)$
$\land\ [lr \mapsto L2,\ q \mapsto WQ] \in TrustLive$
$\land\ \forall\, a \in WQ :$
$\exists\, m2 \in \{mm \in received[A2] : mm.type = \text{“1b”} \land mm.lr = L2 \land mm.bal = B2\} :$
$\land\ m2.acc\ = a$
$\land\ \exists\, p \in m2.proposals :$
$\land\ p.lr = L2$
$\land\ p.bal = c$
$\land\ p.val = V2$
BY $\langle 4\rangle$3b, $\langle 4\rangle$0a DEF *KnowsSafeAt*2, *Ballot*

$\langle 5\rangle$2. PICK $S1 \in SafeAcceptor : S1 \in Q1 \land S1 \in BQ$BY *EntanglementTrustLive*, $\langle 4\rangle$0, $\langle 5\rangle$1

$\langle 5\rangle$4. PICK $m1 \in received[A2]$:
$\land\ m1.type = \text{“1b”}$
$\land\ m1.lr = L2$
$\land\ m1.bal\ = B2$
$\land\ m1.acc = S1$
$\land\ \forall\, p \in \{pp \in m1.votes : \langle pp.lr,\ L2\rangle \in connected[A2]\} :$
$\land\ p.bal \leq c$
$\land\ p.bal = c \Rightarrow p.val = V2$
BY $\langle 5\rangle$1, $\langle 5\rangle$2

$\langle 5\rangle$5. $\land\ m1 \in msgs$
$\land\ m1.type = \text{“1b”}$
$\land\ m1.lr = L2$
$\land\ m1.bal\ = B2$
$\land\ m1.acc = S1$
$\land\ \forall\, p \in \{pp \in m1.votes : \langle pp.lr,\ L2\rangle \in connected[A2]\} :$
$\land\ p.bal \leq c$
$\land\ p.bal = c \Rightarrow p.val = V2$
BY $\langle 5\rangle$4, *SafeAcceptorIsAcceptor* DEF *TypeOK*, *ReceivedSpec*

$\langle 5\rangle$6.CASE $\neg\, VotedFor(L1,\ S1,\ B1,\ V1)$

$\langle 6\rangle$1. $\neg\, VotedFor(L1,\ S1,\ B1,\ V1)'$BY $\langle 5\rangle$6, $\langle 2\rangle$2 DEF *VotedFor*, *Phase*2*av*, *Send*

$\langle 6\rangle$2. QED BY $\langle 6\rangle$1, $\langle 5\rangle$2, $\langle 5\rangle$5, *MsgsMonotone* DEF *LeftBallot*, *CannotDecide*

$\langle 5\rangle$7.CASE $VotedFor(L1,\ S1,\ B1,\ V1)$

$\langle 6\rangle$1. $[lr \mapsto L1,\ bal \mapsto B1,\ val \mapsto V1] \in votesSent[S1]$BY $\langle 5\rangle$7 DEF *VotesSentSpec*2

$\langle 6\rangle$2. PICK $P \in votesSent[S1] : MaxVote(S1,\ B2,\ P) \land P.lr = L1 \land B1 \leq P.bal$

$\langle 7\rangle$1. SUFFICES ASSUME NEW $vote \in votesSent[S1]$, $vote = [lr \mapsto L1,\ bal \mapsto B1,\ val \mapsto V1]$
PROVE $\exists\, P \in votesSent[S1] : MaxVote(S1,\ B2,\ P) \land P.lr = L1 \land vote.bal \leq P.bal$
BY $\langle 6\rangle$1

$\langle 7\rangle$2. QED BY $\langle 7\rangle$1, *SafeAcceptorIsAcceptor* DEF *VotesSentSpec*3, *TypeOK*

$\langle 6\rangle$3. $P \in m1.votes$BY $\langle 6\rangle$2, $\langle 5\rangle$5 DEF *MsgInv*1*b*

$\langle 6\rangle$4. $\langle P.lr,\ L2\rangle \in connected[A2]$BY $\langle 5\rangle$5, $\langle 6\rangle$2 DEF *ConnectedSpec*

$\langle 6\rangle$5. $P.bal \in Ballot$BY $\langle 5\rangle$5, $\langle 6\rangle$3, *SafeAcceptorIsAcceptor*, *MessageType* DEF *TypeOK*

$\langle 6\rangle$6. $B1 < c$

$\langle 7\rangle$1.CASE $P.val = V1$

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

⟨2⟩3.CASE $Phase2b(lrn,\ bal,\ acc,\ val)$
⟨3⟩1. $m \in msgs$BY ⟨2⟩3, ⟨1⟩0b DEF $Phase2b$, $Send$, $TypeOK$
⟨3⟩2. QED BY ⟨3⟩1 DEF $HeterogeneousSpec$
⟨2⟩4. QED BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3.CASE $AcceptorReceiveAction$BY ⟨1⟩3 DEF $AcceptorReceiveAction$, $Next$, $Recv$, $HeterogeneousSpec$
⟨1⟩4.CASE $AcceptorDisconnectAction$BY ⟨1⟩4 DEF $AcceptorDisconnectAction$, $Disconnect$, $Next$, $Heterogeneou$
⟨1⟩5.CASE $LearnerAction$
⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$,
$\lor\ LearnerDecide(lrn,\ bal)$
$\lor\ LearnerRecv(lrn)$
PROVE $CannotDecide(Q1,\ L1,\ B1,\ V1)'$
BY ⟨1⟩5 DEF $LearnerAction$
⟨2⟩2.CASE $LearnerDecide(lrn,\ bal)$BY ⟨2⟩2 DEF $LearnerDecide$, $Next$, $HeterogeneousSpec$
⟨2⟩3.CASE $LearnerRecv(lrn)$BY ⟨2⟩2 DEF $LearnerRecv$, $Next$, $HeterogeneousSpec$
⟨2⟩4. QED BY ⟨2⟩2, ⟨2⟩3
⟨1⟩6.CASE $FakeAcceptorAction$BY ⟨1⟩6, $SafeAcceptorAssumption$ DEF $FakeAcceptorAction$, $FakeSend$, $Send$, H
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF $Next$

LEMMA $ChosenSafeCaseEq \triangleq$
ASSUME NEW $L1 \in Learner$, NEW $L2 \in Learner$,
NEW $B \in Ballot$,
NEW $V1 \in Value$, NEW $V2 \in Value$,
$TypeOK$, $MsgInv$,
$ReceivedSpec$, $ReceivedByLearnerSpec$, $VotesSentSpec4$,
$\langle L1,\ L2\rangle \in Ent$,
$ChosenIn(L1,\ B,\ V1)$, $ChosenIn(L2,\ B,\ V2)$
PROVE $V1 = V2$
PROOF
⟨1⟩ USE DEF $MsgInv$
⟨1⟩1. PICK $Q1 \in ByzQuorum$:
$\land\ [lr \mapsto L1,\ q \mapsto Q1] \in TrustLive$
$\land\ \forall\, aa \in Q1 :$
$\exists\, m \in \{mm \in receivedByLearner[L1] : mm.bal = B\} :$
$\land\ m.val = V1$
$\land\ m.acc = aa$
BY DEF $ChosenIn$
⟨1⟩2. PICK $Q2 \in ByzQuorum$:
$\land\ [lr \mapsto L2,\ q \mapsto Q2] \in TrustLive$
$\land\ \forall\, aa \in Q2 :$
$\exists\, m \in \{mm \in receivedByLearner[L2] : mm.bal = B\} :$
$\land\ m.val = V2$
$\land\ m.acc = aa$
BY DEF $ChosenIn$
⟨1⟩3. PICK $A \in SafeAcceptor : A \in Q1 \land A \in Q2$BY $EntanglementTrustLive$, ⟨1⟩1, ⟨1⟩2
⟨1⟩4. PICK $m1 \in receivedByLearner[L1] : m1.acc = A \land m1.bal = B \land m1.val = V1$BY ⟨1⟩1, ⟨1⟩3 DEF $Chosen$

⟨1⟩5. PICK $m2 \in receivedByLearner[L2] : m2.acc = A \land m2.bal = B \land m2.val = V2$BY ⟨1⟩2, ⟨1⟩3 DEF *Chosen*
⟨1⟩6. $\land\ m1 \in msgs$
$\land\ m1.type =$ "2b"
$\land\ m1.lr = L1$
$\land\ m1.acc = A$
$\land\ m1.bal = B$
$\land\ m1.val = V1$
BY ⟨1⟩4 DEF *ReceivedByLearnerSpec*, *TypeOK*
⟨1⟩7. $\land\ m2 \in msgs$
$\land\ m2.type =$ "2b"
$\land\ m2.lr = L2$
$\land\ m2.acc = A$
$\land\ m2.bal = B$
$\land\ m2.val = V2$
BY ⟨1⟩5 DEF *ReceivedByLearnerSpec*, *TypeOK*
⟨1⟩8. $[lr \mapsto L1,\ bal \mapsto B,\ val \mapsto V1] \in votesSent[A]$BY ⟨1⟩6 DEF *MsgInv*2*b*
⟨1⟩9. $[lr \mapsto L2,\ bal \mapsto B,\ val \mapsto V2] \in votesSent[A]$BY ⟨1⟩7 DEF *MsgInv*2*b*
⟨1⟩100. QED BY ⟨1⟩8, ⟨1⟩9 DEF *VotesSentSpec*4

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

BY DEF *ChosenIn*

⟨1⟩3. PICK $A \in \mathit{SafeAcceptor} : A \in Q1 \land A \in Q2$BY *EntanglementTrustLive*, ⟨1⟩1, ⟨1⟩2

⟨1⟩4. PICK $m1 \in \mathit{receivedByLearner}[L1] : m1.acc = A \land m1.bal = B1 \land m1.val = V1$BY ⟨1⟩1, ⟨1⟩3 DEF *Chose*

⟨1⟩5. PICK $m2 \in \mathit{receivedByLearner}[L2] : m2.acc = A \land m2.bal = B2 \land m2.val = V2$BY ⟨1⟩2, ⟨1⟩3 DEF *Chose*

⟨1⟩6. $\land\ m1 \in \mathit{msgs}$
$\land\ m1.\mathit{type} =$ "2b"
$\land\ m1.lr = L1$
$\land\ m1.acc = A$
$\land\ m1.bal = B1$
$\land\ m1.val = V1$
BY ⟨1⟩4 DEF *ReceivedByLearnerSpec*, *TypeOK*

⟨1⟩7. $\land\ m2 \in \mathit{msgs}$
$\land\ m2.\mathit{type} =$ "2b"
$\land\ m2.lr = L2$
$\land\ m2.acc = A$
$\land\ m2.bal = B2$
$\land\ m2.val = V2$
BY ⟨1⟩5 DEF *ReceivedByLearnerSpec*, *TypeOK*

⟨1⟩10. PICK $R1 \in \mathit{ByzQuorum}$:
$\land\ [lr \mapsto L1,\ q \mapsto R1] \in \mathit{TrustLive}$
BY ⟨1⟩6 DEF *MsgInv2b*

⟨1⟩11. PICK $R2 \in \mathit{ByzQuorum}$:
$\land\ [lr \mapsto L2,\ q \mapsto R2] \in \mathit{TrustLive}$
$\land\ \forall\, aa \in R2 :$
$\exists\, m2av \in \mathit{received}[A] :$
$\land\ m2av.\mathit{type} =$ "2av"
$\land\ m2av.lr = L2$
$\land\ m2av.acc = aa$
$\land\ m2av.bal = B2$
$\land\ m2av.val = V2$
BY ⟨1⟩7 DEF *MsgInv2b*

⟨1⟩12. PICK $A0 \in \mathit{SafeAcceptor} : A0 \in R1 \land A0 \in R2$BY *EntanglementTrustLive*, ⟨1⟩10, ⟨1⟩11

⟨1⟩14. PICK $m2av2 \in \mathit{received}[A]$:
$m2av2.\mathit{type} =$ "2av" $\land\ m2av2.lr = L2 \land m2av2.acc = A0 \land m2av2.bal = B2 \land m2av2.val = V2$
BY ⟨1⟩12, ⟨1⟩11

⟨1⟩16. $\land\ m2av2 \in \mathit{msgs}$
$\land\ m2av2.\mathit{type} =$ "2av"
$\land\ m2av2.lr = L2$
$\land\ m2av2.acc = A0$
$\land\ m2av2.bal = B2$
$\land\ m2av2.val = V2$
BY ⟨1⟩14, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec*, *TypeOK*

⟨1⟩17. *CannotDecide*($Q1$, $L1$, $B1$, $V1$)

⟨2⟩1. $[lr \mapsto L1,\ q \mapsto Q1] \in \mathit{TrustLive}$BY ⟨1⟩1

⟨2⟩5. QED BY ⟨1⟩16, ⟨2⟩1 DEF *HeterogeneousSpec*

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

LEMMA *ChosenSafe* $\triangleq$
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*, *VotesSentSpec*4, *MsgInv*,
HeterogeneousSpec,
$\langle L1,\ L2\rangle \in Ent$,
$ChosenIn(L1,\ B1,\ V1)$, $ChosenIn(L2,\ B2,\ V2)$
PROVE $V1 = V2$
PROOF
⟨1⟩ USE DEF *MsgInv*
⟨1⟩1. PICK $Q1 \in ByzQuorum$:
$\land\ [lr \mapsto L1,\ q \mapsto Q1] \in TrustLive$
$\land\ \forall\, aa \in Q1 :$
$\exists\, m \in \{mm \in receivedByLearner[L1] : mm.bal = B1\} :$
$\land\ m.val = V1$
$\land\ m.acc = aa$
BY DEF *ChosenIn*
⟨1⟩2. PICK $Q2 \in ByzQuorum$:
$\land\ [lr \mapsto L2,\ q \mapsto Q2] \in TrustLive$
$\land\ \forall\, aa \in Q2 :$
$\exists\, m \in \{mm \in receivedByLearner[L2] : mm.bal = B2\} :$
$\land\ m.val = V2$
$\land\ m.acc = aa$
BY DEF *ChosenIn*
⟨1⟩3. PICK $A \in SafeAcceptor : A \in Q1 \land A \in Q2$BY *EntanglementTrustLive*, ⟨1⟩1, ⟨1⟩2
⟨1⟩4. PICK $m1 \in receivedByLearner[L1] : m1.acc = A \land m1.bal = B1 \land m1.val = V1$BY ⟨1⟩1, ⟨1⟩3 DEF *Chose*
⟨1⟩5. PICK $m2 \in receivedByLearner[L2] : m2.acc = A \land m2.bal = B2 \land m2.val = V2$BY ⟨1⟩2, ⟨1⟩3 DEF *Chose*
⟨1⟩6. $\land\ m1 \in msgs$
$\land\ m1.type = \text{“2b”}$
$\land\ m1.lr = L1$
$\land\ m1.acc = A$
$\land\ m1.bal = B1$
$\land\ m1.val = V1$

BY $\langle 1\rangle 4$ DEF *ReceivedByLearnerSpec*, *TypeOK*

$\langle 1\rangle 7$. $\land\ m2 \in msgs$
$\land\ m2.type = \text{“2b”}$
$\land\ m2.lr = L2$
$\land\ m2.acc = A$
$\land\ m2.bal = B2$
$\land\ m2.val = V2$
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*

$\langle 1\rangle 10$. PICK $R1 \in ByzQuorum$:
$\land\ [lr \mapsto L1,\ q \mapsto R1] \in TrustLive$
$\land\ \forall\, aa \in R1$:
$\exists\, m2av \in received[A]$:
$\land\ m2av.type = \text{“2av”}$
$\land\ m2av.lr = L1$
$\land\ m2av.acc = aa$
$\land\ m2av.bal = B1$
$\land\ m2av.val = V1$
BY $\langle 1\rangle 6$ DEF *MsgInv2b*

$\langle 1\rangle 11$. PICK $R2 \in ByzQuorum$:
$\land\ [lr \mapsto L2,\ q \mapsto R2] \in TrustLive$
$\land\ \forall\, aa \in R2$:
$\exists\, m2av \in received[A]$:
$\land\ m2av.type = \text{“2av”}$
$\land\ m2av.lr = L2$
$\land\ m2av.acc = aa$
$\land\ m2av.bal = B2$
$\land\ m2av.val = V2$
BY $\langle 1\rangle 7$ DEF *MsgInv2b*

$\langle 1\rangle 12$. PICK $A0 \in SafeAcceptor : A0 \in R1 \land A0 \in R2$BY *EntanglementTrustLive*, $\langle 1\rangle 10$, $\langle 1\rangle 11$

$\langle 1\rangle 13$. PICK $m2av1 \in received[A]$:
$m2av1.type = \text{“2av”} \land m2av1.lr = L1 \land m2av1.acc = A0 \land m2av1.bal = B1 \land m2av1.val = V1$
BY $\langle 1\rangle 12$, $\langle 1\rangle 10$

$\langle 1\rangle 14$. PICK $m2av2 \in received[A]$:
$m2av2.type = \text{“2av”} \land m2av2.lr = L2 \land m2av2.acc = A0 \land m2av2.bal = B2 \land m2av2.val = V2$
BY $\langle 1\rangle 12$, $\langle 1\rangle 11$

$\langle 1\rangle 15$. $\land\ m2av1 \in msgs$
$\land\ m2av1.type = \text{“2av”}$
$\land\ m2av1.lr = L1$
$\land\ m2av1.acc = A0$
$\land\ m2av1.bal = B1$
$\land\ m2av1.val = V1$
BY $\langle 1\rangle 13$, *SafeAcceptorIsAcceptor* DEF *ReceivedSpec*, *TypeOK*

$\langle 1\rangle 16$. $\land\ m2av2 \in msgs$

$\land\ m2av2.type = \text{“2av”}$
$\land\ m2av2.lr = L2$
$\land\ m2av2.acc = A0$
$\land\ m2av2.bal = B2$
$\land\ m2av2.val = V2$
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*

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

LEMMA $SafetyStep \triangleq$
$TypeOK \land Next \land MsgInv \land$
$DecisionSpec \land ReceivedSpec \land ReceivedByLearnerSpec \land$
$2avSentSpec1 \land 2avSentSpec3 \land VotesSentSpec4 \land$
$HeterogeneousSpec \land Safety \Rightarrow Safety'$
PROOF
$\langle 1\rangle$ SUFFICES
ASSUME *TypeOK*, *Next*, *MsgInv*, *Safety*, *DecisionSpec*, *ReceivedSpec*, *ReceivedByLearnerSpec*,
$2avSentSpec1$, $2avSentSpec3$, *VotesSentSpec*4,
HeterogeneousSpec,
NEW $L1 \in Learner$, NEW $L2 \in Learner$,
NEW $B1 \in Ballot$, NEW $B2 \in Ballot$,
NEW $V1 \in Value$, NEW $V2 \in Value$,
$\langle L1,\ L2\rangle \in Ent$,
$V1 \in decision'[L1,\ B1],\ V2 \in decision'[L2,\ B2]$
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*, *Phase*1*a*, *Phase*1*c*, *Send*, *Safety*
$\langle 1\rangle 2$.CASE *AcceptorSendAction*
$\langle 2\rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,
NEW $bal \in Ballot$,
NEW $acc \in SafeAcceptor$,
NEW $val \in Value$,
$\lor\ Phase1b(lrn,\ bal,\ acc)$
$\lor\ Phase2av(lrn,\ bal,\ acc,\ val)$
$\lor\ Phase2b(lrn,\ bal,\ acc,\ val)$
PROVE $V1 = V2$

BY ⟨1⟩2 DEF *AcceptorSendAction*
⟨2⟩2.CASE *Phase*1*b*(*lrn*, *bal*, *acc*)BY ⟨2⟩2, ⟨1⟩0a, ⟨1⟩0b DEF *AcceptorSendAction*, *Send*, *Phase*1*b*, *Safety*, *Ty*
⟨2⟩3.CASE *Phase*2*av*(*lrn*, *bal*, *acc*, *val*)BY ⟨2⟩3, ⟨1⟩0a, ⟨1⟩0b DEF *AcceptorSendAction*, *Send*, *Phase*2*av*, *Saf*
⟨2⟩4.CASE *Phase*2*b*(*lrn*, *bal*, *acc*, *val*)BY ⟨2⟩4, ⟨1⟩0a, ⟨1⟩0b DEF *AcceptorSendAction*, *Send*, *Phase*2*b*, *Safety*
⟨2⟩5. QED BY ⟨2⟩2, ⟨2⟩3, ⟨2⟩4
⟨1⟩3.CASE *AcceptorReceiveAction*BY ⟨1⟩3, ⟨1⟩0a, ⟨1⟩0b DEF *AcceptorReceiveAction*, *Recv*, *TypeOK*, *Safety*
⟨1⟩4.CASE *AcceptorDisconnectAction*BY ⟨1⟩4 DEF *AcceptorDisconnectAction*, *Disconnect*, *Safety*
⟨1⟩5.CASE *LearnerAction*
⟨2⟩ SUFFICES ASSUME NEW *lrn* ∈ *Learner*, NEW *bal* ∈ *Ballot*,
∨ *LearnerDecide*(*lrn*, *bal*)
∨ *LearnerRecv*(*lrn*)
PROVE $V1 = V2$BY ⟨1⟩5 DEF *LearnerAction*
⟨2⟩1.CASE *LearnerRecv*(*lrn*)BY ⟨2⟩1 DEF *LearnerRecv*, *Safety*
⟨2⟩2.CASE *LearnerDecide*(*lrn*, *bal*)
⟨3⟩ SUFFICES ASSUME NEW *val* ∈ *Value*,
ChosenIn(*lrn*, *bal*, *val*),
$decision' = [decision$ EXCEPT $![\langle lrn, bal\rangle] = decision[lrn, bal] \cup \{val\}]$,
UNCHANGED ⟨*msgs*, *maxBal*, *votesSent*, 2*avSent*, *received*, *connected*, *receivedByLe*
PROVE $V1 = V2$
BY ⟨2⟩2 DEF *LearnerDecide*
⟨3⟩0.CASE $V1 = V2$BY ⟨3⟩0
⟨3⟩1.CASE $V1 \neq V2$
⟨4⟩1.CASE $val \neq V1 \land val \neq V2$BY ⟨4⟩1 DEF *Safety*, *TypeOK*
⟨4⟩2.CASE $val = V1$
⟨5⟩0. $V2 \in decision[L2, B2]$BY ⟨3⟩1, ⟨4⟩2 DEF *TypeOK*
⟨5⟩1. *ChosenIn*(*L*2, *B*2, *V* 2)BY ⟨5⟩0 DEF *DecisionSpec*
⟨5⟩2.CASE $V1 \in decision[L1, B1]$BY ⟨5⟩0, ⟨5⟩2 DEF *Safety*
⟨5⟩3.CASE $V1 \notin decision[L1, B1]$
⟨6⟩1. $lrn = L1 \land bal = B1$BY ⟨5⟩3, ⟨4⟩2 DEF *TypeOK*
⟨6⟩2. *ChosenIn*(*L*1, *B*1, *V* 1)BY ⟨6⟩1, ⟨4⟩2
⟨6⟩3. QED BY ⟨5⟩1, ⟨6⟩2, *ChosenSafe*
⟨5⟩4. QED BY ⟨5⟩2, ⟨5⟩3
⟨4⟩3.CASE $val = V2$
⟨5⟩0. $V1 \in decision[L1, B1]$BY ⟨3⟩1, ⟨4⟩3 DEF *TypeOK*
⟨5⟩1. *ChosenIn*(*L*1, *B*1, *V* 1)BY ⟨5⟩0 DEF *DecisionSpec*
⟨5⟩2.CASE $V2 \in decision[L2, B2]$BY ⟨5⟩0, ⟨5⟩2 DEF *Safety*
⟨5⟩3.CASE $V2 \notin decision[L2, B2]$
⟨6⟩1. $lrn = L2 \land bal = B2$BY ⟨5⟩3, ⟨4⟩3 DEF *TypeOK*
⟨6⟩2. *ChosenIn*(*L*2, *B*2, *V* 2)BY ⟨6⟩1, ⟨4⟩3
⟨6⟩10. QED BY ⟨5⟩1, ⟨6⟩2, *ChosenSafe*
⟨5⟩4. QED BY ⟨5⟩2, ⟨5⟩3
⟨4⟩4. QED BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3
⟨3⟩2. QED BY ⟨3⟩0, ⟨3⟩1
⟨2⟩3. QED BY ⟨2⟩1, ⟨2⟩2
⟨1⟩6.CASE *FakeAcceptorAction*BY ⟨1⟩6 DEF *FakeAcceptorAction*, *FakeSend*, *Send*, *Safety*

⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF *Next*

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

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 $FullSafetyInvariantNext \triangleq FullSafetyInvariant \land [Next]_{vars} \Rightarrow FullSafetyInvariant'$
PROOF
$\langle 1 \rangle$ SUFFICES ASSUME $FullSafetyInvariant$, $[Next]_{vars}$PROVE $FullSafetyInvariant'$OBVIOUS
$\langle 1 \rangle$1.CASE $Next$BY $\langle 1 \rangle$1,
$TypeOKInvariant$, $MsgInvInvariant$,
$2avSentSpec1Invariant$, $2avSentSpec2Invariant$, $2avSentSpec3Invariant$,
$VotesSentSpec1Invariant$, $VotesSentSpec2Invariant$, $VotesSentSpec3Invariant$, $VotesSentSpec4Invarian$
$ReceivedSpecInvariant$, $ReceivedByLearnerSpecInvariant$, $ConnectedSpecInvariant$, $DecisionSpecInvaria$
$HeterogeneousSpecInvariant$, $SafetyStep$
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 $Spec$, $FullSafetyInvariant$