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 $BQAssumption \triangleq$
$\land\ \forall\, Q \in ByzQuorum : Q \subseteq Acceptor$

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

Learner graph

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

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

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

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

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

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

```
LEMMA EntanglementTrustLive ≜
    ASSUME NEW L1 ∈ Learner, NEW L2 ∈ Learner,
           NEW Q1 ∈ ByzQuorum, NEW Q2 ∈ ByzQuorum,
           ⟨L1, L2⟩ ∈ Ent,
           [lr ↦ L1, q ↦ Q1] ∈ TrustLive,
           [lr ↦ L2, q ↦ Q2] ∈ TrustLive
    PROVE  ∃ N ∈ SafeAcceptor : N ∈ Q1 ∧ N ∈ Q2
PROOF BY EntanglementAssumption, LearnerGraphAssumption
```

Messages

```
Message ≜
    [type : {"1a"}, lr : Learner, bal : Ballot] ∪
    [
        type : {"1b"},
        lr   : Learner,
        acc  : Acceptor,
        bal  : Ballot,
        votes : SUBSET [lr : Learner, bal : Ballot, val : Value],
        proposals : SUBSET [lr : Learner, bal : Ballot, val : Value]
    ] ∪
    [type : {"1c"}, lr : Learner, bal : Ballot, val : Value] ∪
    [type : {"2av"}, lr : Learner, acc : Acceptor, bal : Ballot, val : Value] ∪
    [type : {"2b"}, lr : Learner, acc : Acceptor, bal : Ballot, val : Value]
```

Algorithm specification

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

InitializedBallot(lr, bal) ≜
    ∃ m ∈ msgs : m.type = "1a" ∧ m.lr = lr ∧ m.bal = bal

AnnouncedValue(lr, bal, val) ≜
    ∃ m ∈ msgs : m.type = "1c" ∧ m.bal = bal ∧ m.val = val

ChosenIn(lr, bal, v) ≜
    ∃ Q ∈ ByzQuorum :
        ∧ [lr ↦ lr, q ↦ Q] ∈ TrustLive
        $\land \forall aa \in Q :$
            $\exists m \in \{mm \in receivedByLearner[lr] : mm.bal = bal\} :$
                $\land m.val = v$
                $\land m.acc = aa$

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

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

$KnowsSafeAt(l, ac, b, v) \triangleq$
    $\lor KnowsSafeAt1(l, ac, b, v)$
    $\lor KnowsSafeAt2(l, ac, b, v)$

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

$TypeOK \triangleq$
    $\land$  $msgs \in$ SUBSET $Message$
    $\land$  $maxBal \in [Learner \times Acceptor \rightarrow Ballot]$
```

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

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

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

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

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

$$\begin{array}{l}
\mathit{MaxVote}(a,\ b,\ \mathit{vote})\ \triangleq \\
\quad \land\ \ \mathit{vote}.\mathit{bal} < b \\
\quad \land\ \ \forall\, \mathit{other}\ \in \mathit{votesSent}[a] : \\
\quad\qquad \mathit{other}.\mathit{lr} = \mathit{vote}.\mathit{lr} \land \mathit{other}.\mathit{bal} < b \Rightarrow \\
\quad\qquad \mathit{other}.\mathit{bal} \leq \mathit{vote}.\mathit{bal}
\end{array}$$

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

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

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

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

$Disconnect(a) \triangleq$
    $\land\ \exists P \in \textsc{subset}\ \{LL \in Learner \times Learner : LL \notin Ent\} :$
        $connected' = [connected \textsc{ except } ![a] = connected[a] \setminus P]$
    $\land\ \textsc{unchanged}\ \langle msgs, maxBal, votesSent, 2avSent, received, receivedByLearner, decision \rangle$

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

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

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

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

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

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

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

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

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

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

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

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

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

$LeftBallot(lr,\ acc,\ bal)\ \triangleq$
    $\exists\, m \in msgs :$
        $\land\ m.type = \text{“1b”}$
        $\land\ m.lr = lr$
        $\land\ m.acc = acc$
        $\land\ bal < m.bal$
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
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 $Phase1b, Send, TypeOK$
      $\langle 4\rangle 1.$ $\{vote \in votesSent[acc] : MaxVote(acc, bal, vote)\}$
                $\in$ SUBSET $[lr : Learner, bal : Ballot, val : Value]$
            BY  DEF $TypeOK$
      $\langle 4\rangle 2.$ $\{p \in 2avSent[acc] : p.bal < bal \land p.lr = lrn\} \in$ SUBSET $[lr : Learner, bal : Ballot, val : Value]$
            BY  DEF $TypeOK$
      $\langle 4\rangle 3.$ QED BY $\langle 4\rangle 1$, $\langle 4\rangle 2$  DEF $Message$, $TypeOK$
    $\langle 3\rangle 4.$ QED BY $\langle 2\rangle 1$, $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$  DEF $Phase1b$, $TypeOK$, $Send$
  $\langle 2\rangle 2.$CASE $Phase2av(lrn, bal, acc, val)$
    $\langle 3\rangle 2.$ $msgs' \in$ SUBSET $Message$
        $\langle 4\rangle 0.$ $[type \mapsto \text{“2av”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val] \in Message$
            BY $SafeAcceptorIsAcceptor$ DEF $Message$
        $\langle 4\rangle 1.$ QED BY $\langle 2\rangle 2$, $\langle 4\rangle 0$, $SafeAcceptorIsAcceptor$ DEF $Phase2av$, $Send$, $TypeOK$, $Message$
    $\langle 3\rangle 4.$ $(2avSent \in [Acceptor \rightarrow$ SUBSET $[lr : Learner, bal : Ballot, val : Value]])'$
        $\langle 4\rangle 0.$ $[lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \in [lr : Learner, bal$          $: Ballot, val : Value]$
              BY  DEF $TypeOK$
        $\langle 4\rangle 1.$ QED BY $\langle 2\rangle 2$, $\langle 1\rangle 2$, $\langle 4\rangle 0$, $SafeAcceptorIsAcceptor$ DEF $Phase2av$, $Send$, $TypeOK$, $Message$
    $\langle 3\rangle 5.$ QED BY $\langle 2\rangle 2$, $\langle 3\rangle 2$, $\langle 3\rangle 4$  DEF $Phase2av$, $Send$, $TypeOK$
  $\langle 2\rangle 3.$CASE $Phase2b(lrn, bal, acc, val)$
    $\langle 3\rangle 1.$ $val \in Value$OBVIOUS
    $\langle 3\rangle 2.$ $msgs' \in$ SUBSET $Message$
        $\langle 4\rangle 0.$ $[type \mapsto \text{“2b”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val] \in Message$
            BY $SafeAcceptorIsAcceptor$ DEF $Message$
        $\langle 4\rangle 1.$ QED BY $\langle 4\rangle 0$, $\langle 2\rangle 3$  DEF $Phase2b$, $Message$, $Send$, $TypeOK$
    $\langle 3\rangle 3.$ $votesSent' \in [Acceptor \rightarrow$ SUBSET $[lr : Learner, bal : Ballot, val : Value]]$
        $\langle 4\rangle 0.$ $[lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \in [lr : Learner, bal : Ballot, val : Value]$BY $\langle 3\rangle 1$
        $\langle 4\rangle 1$ QED BY $\langle 2\rangle 3$, $\langle 1\rangle 2$, $\langle 4\rangle 0$  DEF $Phase2b$, $TypeOK$
    $\langle 3\rangle 5.$ QED BY $\langle 2\rangle 3$, $\langle 1\rangle 2$, $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$  DEF $Phase2b$, $Send$, $TypeOK$
  $\langle 2\rangle 4.$ QED BY $\langle 1\rangle 2$, $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$
$\langle 1\rangle 3.$CASE $AcceptorReceiveAction$
  $\langle 2\rangle$ SUFFICES ASSUME NEW $lrn \in Learner$,
                          NEW $acc \in Acceptor$,
                          NEW $m \in msgs$,
                          $received' = [received$ EXCEPT $![acc] = received[acc] \cup \{m\}]$,
                          UNCHANGED $\langle msgs, maxBal, 2avSent, votesSent, connected,$
                                      $receivedByLearner, decision\rangle$
               PROVE  $TypeOK'$
    BY $SafeAcceptorIsAcceptor$, $\langle 1\rangle 3$  DEF $AcceptorReceiveAction$, $Recv$
  $\langle 2\rangle 7.$ QED BY $\langle 1\rangle 3$  DEF $AcceptorReceiveAction$, $Recv$, $TypeOK$
$\langle 1\rangle 4.$CASE $AcceptorDisconnectAction$BY $\langle 1\rangle 4$  DEF $AcceptorDisconnectAction$, $Disconnect$, $TypeOK$, $Message$
$\langle 1\rangle 5.$CASE $LearnerAction$
  $\langle 2\rangle 1.$ ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$,
                 $LearnerDecide(lrn, bal)$
                 PROVE  $TypeOK'$
    BY $\langle 2\rangle 1$  DEF $LearnerDecide$, $TypeOK$
  $\langle 2\rangle 2.$ 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 EXCEPT ![acc] = received[acc] ∪ {m}],
                        UNCHANGED ⟨msgs, maxBal, 2avSent, votesSent, connected, receivedByLearner, decis
                 PROVE  ReceivedSpec′
    BY ⟨1⟩3, SafeAcceptorIsAcceptor DEF AcceptorReceiveAction, Recv
  ⟨2⟩ QED BY MessageType, SafeAcceptorIsAcceptor DEF ReceivedSpec, TypeOK, Next
⟨1⟩4.CASE AcceptorDisconnectAction
  BY ⟨1⟩4  DEF AcceptorDisconnectAction, Disconnect, ReceivedSpec, TypeOK, Next
⟨1⟩5.CASE LearnerAction
  BY ⟨1⟩5  DEF LearnerAction, LearnerRecv, LearnerDecide, ReceivedSpec, TypeOK, Next
⟨1⟩6.CASE FakeAcceptorAction
  ⟨2⟩1. SUFFICES ASSUME NEW a ∈ Acceptor, FakeSend(a)PROVE ReceivedSpec′
        BY ⟨1⟩6, FakeAcceptorIsAcceptor DEF FakeAcceptorAction
  ⟨2⟩2. QED BY ⟨2⟩1  DEF FakeSend, Send, TypeOK, ReceivedSpec
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6  DEF Next

LEMMA ReceivedByLearnerSpecInvariant ≜
    TypeOK ∧ ReceivedByLearnerSpec ∧ Next ⇒ ReceivedByLearnerSpec′
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, ReceivedByLearnerSpec, NextPROVE ReceivedByLearnerSpec′OBVIOUS
⟨1⟩1.CASE ProposerAction
  BY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, ReceivedByLearnerSpec, Send, Next, TypeOK
⟨1⟩2.CASE AcceptorSendAction
  ⟨2⟩ SUFFICES ASSUME NEW lrn ∈ Learner,
                      NEW bal ∈ Ballot,
                      NEW acc ∈ Acceptor,
                      NEW val ∈ Value,
                      ∨ Phase1b(lrn, bal, acc)
                      ∨ Phase2av(lrn, bal, acc, val)
                      ∨ Phase2b(lrn, bal, acc, val)
               PROVE  ReceivedByLearnerSpec′
    BY ⟨1⟩2, SafeAcceptorIsAcceptor DEF AcceptorSendAction
  ⟨2⟩1.CASE Phase1b(lrn, bal, acc)
    BY ⟨2⟩1  DEF TypeOK, ReceivedByLearnerSpec, Phase1b, Send
  ⟨2⟩2.CASE Phase2av(lrn, bal, acc, val)
    BY ⟨2⟩2  DEF TypeOK, ReceivedByLearnerSpec, Phase2av, Send
  ⟨2⟩3.CASE Phase2b(lrn, bal, acc, val)
    ⟨3⟩ SUFFICES ASSUME Send([type ↦ "2b", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val])
                 PROVE  ReceivedByLearnerSpec′
      BY ⟨2⟩3  DEF Phase2b
    ⟨3⟩0. TypeOK′BY TypeOKInvariant
    ⟨3⟩1. UNCHANGED ⟨receivedByLearner⟩BY ⟨2⟩3  DEF Phase2b
    ⟨3⟩3. (∀ L ∈ Learner : ∀ mm ∈ Message : mm ∈ receivedByLearner[L] ⇒ mm.lr = L)′
          BY ⟨3⟩1  DEF ReceivedByLearnerSpec, TypeOK
    ⟨3⟩4. (receivedByLearner ∈ [Learner → SUBSET {mm ∈ msgs : mm.type = "2b"}])′
          BY ⟨3⟩0, ⟨3⟩1, MessageType DEF ReceivedByLearnerSpec, Send, TypeOK
    $\langle 3\rangle 5.$ QED BY $\langle 3\rangle 3$, $\langle 3\rangle 4$  DEF $ReceivedByLearnerSpec$
  $\langle 2\rangle 4.$ QED BY $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$
$\langle 1\rangle 3.$CASE $AcceptorReceiveAction$
  BY $\langle 1\rangle 3$  DEF $AcceptorReceiveAction$, $Recv$, $ReceivedByLearnerSpec$, $TypeOK$, $Next$
$\langle 1\rangle 4.$CASE $AcceptorDisconnectAction$
  BY $\langle 1\rangle 4$  DEF $AcceptorDisconnectAction$, $Disconnect$, $ReceivedByLearnerSpec$, $TypeOK$, $Next$
$\langle 1\rangle 5.$CASE $LearnerAction$
  $\langle 2\rangle 1.$ ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$, $LearnerDecide(lrn, bal)$
        PROVE  $ReceivedByLearnerSpec'$
    BY $\langle 2\rangle 1$  DEF $LearnerDecide$, $ReceivedByLearnerSpec$, $TypeOK$, $Next$
  $\langle 2\rangle 2.$ ASSUME NEW $lrn \in Learner$, $LearnerRecv(lrn)$
        PROVE  $ReceivedByLearnerSpec'$
    $\langle 3\rangle$ SUFFICES ASSUME NEW $m \in \{mm \in msgs : mm.type = \text{“2b”} \land mm.lr = lrn\}$,
                          $receivedByLearner' =$
                             $[receivedByLearner$ EXCEPT $![lrn] = receivedByLearner[lrn] \cup \{m\}]$
                 PROVE  $ReceivedByLearnerSpec'$
      BY $\langle 2\rangle 2$  DEF $LearnerRecv$
    $\langle 3\rangle 1.$ UNCHANGED $\langle msgs\rangle$BY $\langle 2\rangle 2$  DEF $LearnerAction$, $LearnerRecv$
    $\langle 3\rangle 5.$ QED BY $\langle 2\rangle 2$, $\langle 3\rangle 1$  DEF $ReceivedByLearnerSpec$
  $\langle 2\rangle 3.$ QED BY $\langle 1\rangle 5$, $\langle 2\rangle 1$, $\langle 2\rangle 2$  DEF $LearnerAction$
$\langle 1\rangle 6.$CASE $FakeAcceptorAction$
  $\langle 2\rangle 1.$ SUFFICES ASSUME NEW $a \in Acceptor$, $FakeSend(a)$PROVE  $ReceivedByLearnerSpec'$
        BY $\langle 1\rangle 6$, $FakeAcceptorIsAcceptor$ DEF $FakeAcceptorAction$
  $\langle 2\rangle 2.$ QED BY $\langle 2\rangle 1$  DEF $FakeSend$, $Send$, $TypeOK$, $ReceivedByLearnerSpec$
$\langle 1\rangle 7.$ QED BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$, $\langle 1\rangle 4$, $\langle 1\rangle 5$, $\langle 1\rangle 6$  DEF $Next$

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

```
      ⟨3⟩1.CASE ⟨l, a⟩ = ⟨lrn, acc⟩BY ⟨2⟩1, ⟨3⟩1  DEF Phase1b, TypeOK, Ballot
      ⟨3⟩2.CASE ⟨l, a⟩ ≠ ⟨lrn, acc⟩BY ⟨2⟩1, ⟨3⟩2, SafeAcceptorIsAcceptor DEF Phase1b, TypeOK, Ballot
      ⟨3⟩3. QED BY ⟨3⟩1, ⟨3⟩2
    ⟨2⟩2.CASE Phase2av(lrn, bal, acc, val)
      ⟨3⟩1. UNCHANGED maxBalBY ⟨2⟩2  DEF Phase2av
      ⟨3⟩2. QED BY ⟨3⟩1, SafeAcceptorIsAcceptor DEF TypeOK, Ballot
    ⟨2⟩3.CASE Phase2b(lrn, bal, acc, val)
      ⟨3⟩1. UNCHANGED maxBalBY ⟨2⟩3  DEF Phase2b
      ⟨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 maxBalBY ⟨1⟩3  DEF AcceptorReceiveAction, Recv
    ⟨2⟩2. QED BY ⟨2⟩1, SafeAcceptorIsAcceptor DEF TypeOK, Ballot
  ⟨1⟩4.CASE AcceptorDisconnectAction
    ⟨2⟩1. UNCHANGED maxBalBY ⟨1⟩4  DEF AcceptorDisconnectAction, Disconnect
    ⟨2⟩2. QED BY ⟨2⟩1, SafeAcceptorIsAcceptor DEF TypeOK, Ballot
  ⟨1⟩5.CASE LearnerAction
    ⟨2⟩1. UNCHANGED maxBalBY ⟨1⟩5  DEF LearnerAction, LearnerDecide, LearnerRecv
    ⟨2⟩2. QED BY ⟨2⟩1, SafeAcceptorIsAcceptor DEF TypeOK, Ballot
  ⟨1⟩6.CASE FakeAcceptorAction
    ⟨2⟩1. UNCHANGED maxBalBY ⟨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 ≜ TypeOK ∧ Next ⇒ ∀ A ∈ SafeAcceptor : 2avSent[A] ⊆ 2avSent′[A]
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, NEW A ∈ SafeAcceptorPROVE 2avSent[A] ⊆ 2avSent[A]′OBVIOUS
⟨1⟩0a. TypeOKOBVIOUS
⟨1⟩0b. TypeOK′BY TypeOKInvariant
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Send
⟨1⟩2.CASE AcceptorSendAction
  ⟨2⟩ SUFFICES ASSUME NEW lrn ∈ Learner,
                      NEW bal ∈ Ballot,
                      NEW acc ∈ Acceptor,
                      NEW val ∈ Value,
                      ∨ Phase1b(lrn, bal, acc)
                      ∨ Phase2av(lrn, bal, acc, val)
                      ∨ Phase2b(lrn, bal, acc, val)
               PROVE  2avSent[A] ⊆ 2avSent[A]′
     BY ⟨1⟩2, SafeAcceptorIsAcceptor DEF AcceptorSendAction
  ⟨2⟩1. QED BY ⟨1⟩0b, SafeAcceptorIsAcceptor DEF AcceptorSendAction, Phase1b, Phase2av, Phase2b, Send,
⟨1⟩3.CASE AcceptorReceiveActionBY ⟨1⟩3  DEF AcceptorReceiveAction, Recv
⟨1⟩4.CASE AcceptorDisconnectActionBY ⟨1⟩4  DEF AcceptorDisconnectAction, Disconnect
⟨1⟩5.CASE LearnerActionBY ⟨1⟩5  DEF LearnerAction, LearnerDecide, LearnerRecv
⟨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 ReceivedMonotone ≜
    TypeOK ∧ Next ⇒ ∀ A ∈ SafeAcceptor : received[A] ⊆ received'[A]
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, NEW A ∈ SafeAcceptor
             PROVE  received[A] ⊆ received'[A]OBVIOUS
⟨1⟩0a. TypeOKOBVIOUS
⟨1⟩0b. TypeOK'BY TypeOKInvariant
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Send
⟨1⟩2.CASE AcceptorSendActionBY ⟨1⟩2  DEF AcceptorSendAction, Send, Phase1b, Phase2av, Phase2b
⟨1⟩3.CASE AcceptorReceiveActionBY ⟨1⟩3, ⟨1⟩0a, ⟨1⟩0b, SafeAcceptorIsAcceptor DEF AcceptorReceiveAction, R
⟨1⟩4.CASE AcceptorDisconnectActionBY ⟨1⟩4  DEF AcceptorDisconnectAction, Disconnect
⟨1⟩5.CASE LearnerActionBY ⟨1⟩5  DEF LearnerAction, LearnerDecide, LearnerRecv
⟨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 VotesSentMonotone ≜
    TypeOK ∧ Next ⇒ ∀ A ∈ Acceptor : votesSent[A] ⊆ votesSent'[A]
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, NEW A ∈ AcceptorPROVE  votesSent[A] ⊆ votesSent'[A]OBVIOUS
⟨1⟩0a. TypeOKOBVIOUS
⟨1⟩0b. TypeOK'BY TypeOKInvariant
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Send
⟨1⟩2.CASE AcceptorSendActionBY ⟨1⟩2, ⟨1⟩0a, ⟨1⟩0b  DEF AcceptorSendAction, Send, Phase1b, Phase2av, Ph
⟨1⟩3.CASE AcceptorReceiveActionBY ⟨1⟩3, ⟨1⟩0a, ⟨1⟩0b  DEF AcceptorReceiveAction, Recv, TypeOK
⟨1⟩4.CASE AcceptorDisconnectActionBY ⟨1⟩4  DEF AcceptorDisconnectAction, Disconnect
⟨1⟩5.CASE LearnerActionBY ⟨1⟩5  DEF LearnerAction, LearnerDecide, LearnerRecv
⟨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 VotesSentSpec1Invariant ≜ Next ∧ VotesSentSpec1 ⇒ VotesSentSpec1'
PROOF
⟨1⟩ SUFFICES ASSUME
  Next, VotesSentSpec1, NEW A ∈ SafeAcceptor, NEW vote ∈ votesSent'[A]
    PROVE VotedFor(vote.lr, A, vote.bal, vote.val)'
    BY  DEF VotesSentSpec1
⟨1⟩ USE  DEF VotesSentSpec1
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1, SafeAcceptorIsAcceptor DEF ProposerAction, Phase1a, Phase1c, Next, Se
⟨1⟩2.CASE AcceptorSendAction
  ⟨2⟩.SUFFICES ASSUME NEW lrn ∈ Learner,
                      NEW bal ∈ Ballot,
                      NEW acc ∈ SafeAcceptor,
                      NEW val ∈ Value,
                       ∨ Phase1b(lrn, bal, acc)
                       ∨ Phase2av(lrn, bal, acc, val)
```

```
                               ∨ Phase2b(lrn, bal, acc, val)
                  PROVE   VotedFor(vote.lr, A, vote.bal, vote.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)BY ⟨2⟩2  DEF Phase2av
   ⟨2⟩3.CASE Phase2b(lrn, bal, acc, val)
     ⟨3⟩ SUFFICES ASSUME Send([type ↦ "2b", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val]),
                         votesSent′ = [votesSent EXCEPT ![acc] =
                                          votesSent[acc] ∪ {[lr ↦ lrn, bal ↦ bal, val ↦ val]}]
                    PROVE VotedFor(vote.lr, A, vote.bal, vote.val)′
        BY ⟨2⟩3  DEF Phase2b
     ⟨3⟩2.CASE acc = A
       ⟨4⟩1. USE  DEF VotedFor
       ⟨4⟩2.CASE vote ∈ votesSent[acc]BY ⟨3⟩2, ⟨4⟩2, MsgsMonotone
       ⟨4⟩3.CASE vote ∉ votesSent[acc]
         ⟨5⟩1. DEFINE m0 ≜ [type ↦ "2b", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val]
         ⟨5⟩2. m0 ∈ msgs′BY  DEF Phase2b, Send
         ⟨5⟩3. WITNESS ⟨5⟩2
         ⟨5⟩10 QED BY ⟨3⟩2, ⟨4⟩3
       ⟨4⟩4. QED BY ⟨4⟩2, ⟨4⟩3
     ⟨3⟩3.CASE acc ≠ ABY ⟨3⟩3
     ⟨3⟩4 QED BY ⟨3⟩2, ⟨3⟩3
   ⟨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 VotesSentSpec2Invariant ≜ TypeOK ∧ Next ∧ VotesSentSpec2 ⇒ VotesSentSpec2′
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, VotesSentSpec2,
                    NEW L ∈ Learner, NEW A ∈ SafeAcceptor, NEW B ∈ Ballot, NEW V ∈ Value
             PROVE (VotedFor(L, A, B, V) ⇒ [lr ↦ L, bal ↦ B, val ↦ V] ∈ votesSent[A])′
    BY  DEF VotesSentSpec2
⟨1⟩ USE  DEF VotesSentSpec2
⟨1⟩0a. TypeOKOBVIOUS
⟨1⟩0b. TypeOK′BY TypeOKInvariant
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Send
⟨1⟩2.CASE AcceptorSendAction
   ⟨2⟩.SUFFICES ASSUME NEW lrn ∈ Learner,
                       NEW bal ∈ Ballot,
                       NEW acc ∈ SafeAcceptor,
                       NEW val ∈ Value,
                        ∨ Phase1b(lrn, bal, acc)
```

```
                          ∨ Phase2av(lrn, bal, acc, val)
                          ∨ Phase2b(lrn, bal, acc, val)
                 PROVE  (VotedFor(L, A, B, V) ⇒ [lr ↦ L, bal ↦ B, val ↦ V] ∈ votesSent[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)BY ⟨2⟩2  DEF Phase2av
  ⟨2⟩3.CASE Phase2b(lrn, bal, acc, val)
    ⟨3⟩ SUFFICES ASSUME Send([type ↦ "2b", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val]),
                        votesSent′ = [votesSent EXCEPT ![acc] =
                                      votesSent[acc] ∪ {[lr ↦ lrn, bal ↦ bal, val ↦ val]}]
                   PROVE  (VotedFor(L, A, B, V) ⇒ [lr ↦ L, bal ↦ B, val ↦ V] ∈ votesSent[A])′
        BY ⟨2⟩3  DEF Phase2b
    ⟨3⟩1. QED BY ⟨1⟩0b  DEF Send, VotedFor, TypeOK
  ⟨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 VotesSentSpec3Invariant ≜ TypeOK ∧ Next ∧ VotesSentSpec3 ⇒ VotesSentSpec3′
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, VotesSentSpec3,
                    NEW A ∈ SafeAcceptor, NEW B ∈ Ballot,
                    NEW V ∈ votesSent′[A],
                    V.bal   < B
             PROVE  (∃ P ∈ votesSent[A] : MaxVote(A, B, P) ∧ P.lr = V.lr ∧ V.bal ≤ P.bal)′
    BY  DEF VotesSentSpec3
⟨1⟩ USE  DEF VotesSentSpec3
⟨1⟩0a. TypeOKOBVIOUS
⟨1⟩0b. TypeOK′BY TypeOKInvariant
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Send
⟨1⟩2.CASE AcceptorSendAction
  ⟨2⟩.SUFFICES ASSUME NEW lrn ∈ Learner,
                      NEW bal ∈ Ballot,
                      NEW acc ∈ SafeAcceptor,
                      NEW val ∈ Value,
                      ∨ Phase1b(lrn, bal, acc)
                      ∨ Phase2av(lrn, bal, acc, val)
                      ∨ Phase2b(lrn, bal, acc, val)
               PROVE  (∃ P ∈ votesSent[A] : MaxVote(A, B, P) ∧ P.lr = V.lr ∧ V.bal ≤ P.bal)′
      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)BY ⟨2⟩2  DEF Phase2av
  ⟨2⟩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],
                        ⟨vote1.lr, vote2.lr⟩ ∈ Ent,
                        vote1.bal = vote2.bal
                 PROVE  vote1.val = vote2.val
      BY  DEF  VotesSentSpec4
⟨1⟩ USE  DEF  MsgInv
⟨1⟩0a. TypeOK OBVIOUS
⟨1⟩0b. TypeOK′BY  TypeOKInvariant
⟨1⟩1.CASE  ProposerActionBY  ⟨1⟩1  DEF  ProposerAction, Phase1a, Phase1c, Send, VotesSentSpec4
⟨1⟩2.CASE  AcceptorSendAction
  ⟨2⟩.SUFFICES  ASSUME  NEW  lrn ∈ Learner,
                        NEW  bal ∈ Ballot,
                        NEW  acc ∈ SafeAcceptor,
                        NEW  val ∈ Value,
                        ∨ Phase1b(lrn, bal, acc)
                        ∨ Phase2av(lrn, bal, acc, val)
                        ∨ Phase2b(lrn, bal, acc, val)
                  PROVE  vote1.val = vote2.val
      BY  ⟨1⟩2  DEF  AcceptorSendAction
  ⟨2⟩1.CASE  Phase1b(lrn, bal, acc)BY  ⟨2⟩1  DEF  Phase1b, VotesSentSpec4
  ⟨2⟩2.CASE  Phase2av(lrn, bal, acc, val)BY  ⟨2⟩2  DEF  Phase2av, VotesSentSpec4
  ⟨2⟩3.CASE  Phase2b(lrn, bal, acc, val)
    ⟨3⟩ SUFFICES  ASSUME  votesSent′ = [votesSent  EXCEPT  ![acc] =
                                  votesSent[acc] ∪ {[lr ↦ lrn, bal ↦ bal, val ↦ val]}]
                   PROVE  vote1.val = vote2.val
        BY  ⟨2⟩3  DEF  Phase2b
    ⟨3⟩1.CASE  A = acc
      ⟨4⟩1.CASE  vote1 ∈ votesSent[A] ∧ vote2 ∈ votesSent[A]BY  ⟨4⟩1  DEF  VotesSentSpec4
      ⟨4⟩2.CASE  vote1 ∈ votesSent[A] ∧ vote2 ∉ votesSent[A]
        ⟨5⟩0. vote1.lr ∈ Learner ∧ vote1.val ∈ ValueBY  ⟨4⟩2  DEF  TypeOK
        ⟨5⟩1. vote2 = [lr ↦ lrn, bal ↦ bal, val ↦ val]BY  ⟨4⟩2
        ⟨5⟩2. PICK  Q2 ∈ ByzQuorum :
                 ∧  [lr ↦ lrn, q ↦ Q2] ∈ TrustLive
                 ∧  ∀ aa ∈ Q2 :
                      ∃ m ∈ {mm ∈ received[acc] :
                               ∧ mm.type = "2av"
                               ∧ mm.lr = lrn
                               ∧ mm.bal = bal} :
                          ∧ m.val = val
                          ∧ m.acc = aa
              BY  ⟨5⟩1, ⟨2⟩3  DEF  Phase2b
        ⟨5⟩3. ⟨vote1.lr, lrn⟩ ∈ Ent ∧ vote1.bal = balBY  ⟨5⟩1
        ⟨5⟩4. PICK  m1 ∈ msgs :
                 ∧  m1.type = "2b"
                 ∧  m1.lr = vote1.lr
                 ∧  m1.acc = A
```

```
            $\land\ m1.bal = bal$
            $\land\ m1.val = vote1.val$
          BY $\langle 4\rangle 2$, $\langle 5\rangle 3$  DEF $VotesSentSpec1$, $VotedFor$
$\langle 5\rangle 5$. 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 $MsgInv2b$, $TypeOK$
  $\langle 6\rangle 2$. QED BY $\langle 5\rangle 4$, $\langle 6\rangle 1$, $\langle 3\rangle 1$
$\langle 5\rangle 6$. $\langle vote1.lr,\ lrn\rangle \in Ent$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 $2avSentSpec2$, $Proposed$
$\langle 5\rangle 11$. 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 $\langle 5\rangle 7$, $\langle 5\rangle 2$
      $\langle 5\rangle 12.$ $\land\ m2av2 \in msgs$
            $\land\ m2av2.type =$ "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 $Q1 \in ByzQuorum :$
             $\land$  $[lr \mapsto lrn, q \mapsto Q1] \in TrustLive$
             $\land$  $\forall\, aa \in Q1 :$
                   $\exists\, m \in \{mm \in received[acc] :$
                               $\land\ mm.type =$ "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 $m2 \in msgs :$
             $\land$  $m2.type =$ "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 $VotesSentSpec1$, $VotedFor$
      $\langle 5\rangle 5.$ PICK $Q2 \in ByzQuorum :$
             $\land$  $[lr \mapsto vote2.lr, q \mapsto Q2] \in TrustLive$
             $\land$  $\forall\, ba \in Q2 :$
                   $\exists\, m2av \in received[acc] :$
                     $\land\ m2av.type =$ "2av"
                     $\land\ m2av.lr = vote2.lr$
                     $\land\ m2av.acc = ba$
                     $\land\ m2av.bal = bal$
                     $\land\ m2av.val = vote2.val$
        $\langle 6\rangle 1.$ $\exists\, Q2 \in ByzQuorum :$
```

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

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

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

```
          ⟨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 $Send([type \mapsto \text{"2av"}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val])$,
                        $2avSent' = [2avSent$ EXCEPT $![acc] =$
                                    $2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
                 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 $ProposerAction$, $Phase1a$, $Phase1c$, $Next$, $Send$
⟨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 $V1 = V2$
      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 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 $Phase2av$
    ⟨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 $\land [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \notin 2avSent[A]$
```

```
                $\land [lr \mapsto L2,\ bal \mapsto B,\ val \mapsto V2] \notin 2avSent[A]$
          BY ⟨4⟩4, ⟨3⟩1
      ⟨4⟩5. QED BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3, ⟨4⟩4
    ⟨3⟩2.CASE $A \neq acc$BY ⟨3⟩2
    ⟨3⟩3. QED BY ⟨3⟩1, ⟨3⟩2
  ⟨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 $DecisionSpecInvariant \triangleq TypeOK \land Next \land DecisionSpec \Rightarrow DecisionSpec'$
PROOF
⟨1⟩ 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 $DecisionSpec$
⟨1⟩ USE  DEF $DecisionSpec$
⟨1⟩1.CASE $ProposerAction$BY ⟨1⟩1  DEF $ProposerAction$, $Phase1a$, $Phase1c$, $Next$, $Send$
⟨1⟩2.CASE $AcceptorSendAction$BY ⟨1⟩2  DEF $AcceptorSendAction$, $Phase1b$, $Phase2av$, $Phase2b$, $Next$, $Send$
⟨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 $LearnerAction$
  ⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$,
                      $\lor LearnerDecide(lrn,\ bal)$
                      $\lor LearnerRecv(lrn)$
               PROVE $ChosenIn(L,\ B,\ V)'$
        BY ⟨1⟩5  DEF $LearnerAction$
  ⟨2⟩2.CASE $LearnerDecide(lrn,\ bal)$
    ⟨3⟩0a. $TypeOK$OBVIOUS
    ⟨3⟩0b. $TypeOK'$BY $TypeOKInvariant$
    ⟨3⟩1.CASE $V \in decision[L,\ B]$BY ⟨3⟩1, ⟨2⟩2  DEF $ChosenIn$, $LearnerDecide$
    ⟨3⟩2.CASE $V \notin decision[L,\ B]$BY ⟨3⟩2, ⟨2⟩2, ⟨3⟩0a, ⟨3⟩0b  DEF $ChosenIn$, $LearnerDecide$, $TypeOK$
    ⟨3⟩3. QED BY ⟨3⟩1, ⟨3⟩2
  ⟨2⟩3.CASE $LearnerRecv(lrn)$
    ⟨3⟩1. QED BY ⟨2⟩3  DEF $LearnerRecv$
  ⟨2⟩4. QED BY ⟨2⟩2, ⟨2⟩3  DEF $LearnerAction$
⟨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 $ConnectedSpecInvariant \triangleq Next \land ConnectedSpec \Rightarrow ConnectedSpec'$
PROOF
```

```
⟨1⟩ SUFFICES ASSUME Next, ConnectedSpec,
                      NEW A ∈ SafeAcceptor,
                      NEW L1 ∈ Learner, NEW L2 ∈ Learner,
                      ⟨L1, L2⟩ ∈ Ent
              PROVE ⟨L1, L2⟩ ∈ connected′[A]
    BY DEF ConnectedSpec
⟨1⟩ USE DEF ConnectedSpec
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1 DEF ProposerAction, Phase1a, Phase1c, Next
⟨1⟩2.CASE AcceptorSendActionBY ⟨1⟩2 DEF AcceptorSendAction, Phase1b, Phase2b, Phase2av, Next
⟨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 MsgInvInvariant ≜
    TypeOK ∧ MsgInv ∧ VotesSentSpec1 ∧ VotesSentSpec2 ∧ VotesSentSpec3 ∧ 2avSentSpec1 ∧
    Next ⇒ MsgInv′
PROOF
⟨1⟩ USE DEF MsgInv
⟨1⟩1b. ASSUME TypeOK, VotesSentSpec1, VotesSentSpec2, VotesSentSpec3, 2avSentSpec1, Next,
        ∀ m ∈ msgs : m.acc ∈ SafeAcceptor ∧ m.type = "1b" ⇒ MsgInv1b(m),
        NEW m ∈ msgs′, m.acc ∈ SafeAcceptor, m.type = "1b"
        PROVE MsgInv1b(m)′
  ⟨2⟩0. TypeOKBY ⟨1⟩1b
  ⟨2⟩0a. TypeOK′BY ⟨1⟩1b, TypeOKInvariant
  ⟨2⟩0b. m ∈ MessageBY ⟨2⟩0a DEF TypeOK
  ⟨2⟩0c. maxBal ∈ [Learner × Acceptor → Ballot]BY ⟨1⟩1b DEF TypeOK
  ⟨2⟩0d. maxBal′ ∈ [Learner × Acceptor → Ballot]BY ⟨2⟩0a DEF TypeOK
  ⟨2⟩0e. m.type = "1b"BY ⟨1⟩1b
  ⟨2⟩0f. m.bal ∈ BallotBY ⟨2⟩0b, ⟨2⟩0e DEF Message, Ballot
  ⟨2⟩0g. maxBal[m.lr, m.acc] ∈ BallotBY ⟨2⟩0b, ⟨2⟩0c, ⟨2⟩0e DEF Message
  ⟨2⟩0h. maxBal′[m.lr, m.acc] ∈ BallotBY ⟨2⟩0b, ⟨2⟩0d, ⟨2⟩0e DEF Message
  ⟨2⟩0i. maxBal[m.lr, m.acc] ≤ maxBal′[m.lr, m.acc]BY ⟨1⟩1b, ⟨2⟩0b, MaxBalMonotone DEF TypeOK, Messa
  ⟨2⟩1.CASE ProposerAction
    ⟨3⟩ SUFFICES ASSUME NEW lrn ∈ Learner, NEW proposer ∈ Ballot, NEW val ∈ Value,
                        ∨ Phase1a(lrn, proposer)
                        ∨ Phase1c(lrn, proposer, val)
                  PROVE MsgInv1b(m)′
        BY ⟨2⟩1, ValueNotEmpty DEF ProposerAction
    ⟨3⟩1.CASE Phase1a(lrn, proposer)
      ⟨4⟩1. m ∈ msgsBY ⟨3⟩1, ⟨2⟩0e DEF Phase1a, Send
      ⟨4⟩2. QED BY ⟨1⟩1b, ⟨4⟩1, ⟨3⟩1 DEF Phase1a, MsgInv1b
    ⟨3⟩2.CASE Phase1c(lrn, proposer, val)
      ⟨4⟩1. m ∈ msgsBY ⟨3⟩2, ⟨2⟩0e DEF Phase1c, 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
  $\langle 4\rangle 10.$ QED BY $\langle 4\rangle 1, \langle 4\rangle 5, \langle 4\rangle 6$  DEF $MsgInv1b$
$\langle 3\rangle 2.$CASE $Phase2av(lrn, bal, acc, val)$
  $\langle 4\rangle$ SUFFICES
        ASSUME $maxBal[lrn, acc] \leq bal,$
               $Send([type \mapsto \text{“2av”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]),$
               $2avSent' = [2avSent$ EXCEPT $![acc] =$
                          $2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
               PROVE $MsgInv1b(m)'$
        BY $\langle 3\rangle 2$  DEF $Phase2av$
  $\langle 4\rangle 1.$ $m \in msgs$BY $\langle 2\rangle 0e$  DEF $Send$
  $\langle 4\rangle 1a.$ $m.acc \in Acceptor$BY $\langle 4\rangle 1, MessageType, \langle 2\rangle 0e, \langle 2\rangle 0$  DEF $TypeOK$
  $\langle 4\rangle 2.$ $(m.bal \ \leq maxBal[m.lr, m.acc])'$BY $\langle 1\rangle 1b, \langle 4\rangle 1, \langle 3\rangle 2$  DEF $Phase2av, Send, MsgInv1b$
  $\langle 4\rangle 4.$ $(m.votes = \{p \in votesSent[m.acc] : MaxVote(m.acc, m.bal, p)\})'$
        BY $\langle 1\rangle 1b, \langle 4\rangle 1, \langle 3\rangle 2$  DEF $Phase2av, Send, MsgInv1b$
  $\langle 4\rangle 5.$ $(m.proposals = \{p \in 2avSent[m.acc] : p.bal < m.bal \land p.lr = m.lr\})'$
    $\langle 5\rangle 1.$CASE $m.acc \neq acc$
      $\langle 6\rangle 1.$ $2avSent'[m.acc] = 2avSent[m.acc]$BY $\langle 3\rangle 2, \langle 4\rangle 1, \langle 5\rangle 1, \langle 2\rangle 0, \langle 2\rangle 0e, MessageType$ DEF $Phase2b,$
      $\langle 6\rangle 2.$ QED BY $\langle 6\rangle 1, \langle 4\rangle 1, \langle 1\rangle 1b, \langle 2\rangle 0e$  DEF $MsgInv1b$
    $\langle 5\rangle 2.$CASE $m.acc = acc$
      $\langle 6\rangle 1.$ $m.bal \leq maxBal[m.lr, m.acc]$BY $\langle 1\rangle 1b, \langle 4\rangle 1$  DEF $MsgInv1b$
      $\langle 6\rangle 3.$ $m.bal \in Ballot$BY $\langle 2\rangle 0a, MessageType$ DEF $TypeOK$
      $\langle 6\rangle 5.$ SUFFICES $\{p \in 2avSent[acc] : p.bal < m.bal \land p.lr = m.lr\} =$
                      $\{p \in 2avSent'[acc] : p.bal < m.bal \land p.lr = m.lr\}$
            BY $\langle 4\rangle 1, \langle 2\rangle 0e, \langle 1\rangle 1b, \langle 5\rangle 2$  DEF $MsgInv1b$
      $\langle 6\rangle 6.$ SUFFICES ASSUME NEW $p \in 2avSent'[acc], p.bal < m.bal, p.lr = m.lr$
                      PROVE $p \in 2avSent[acc]$BY $\langle 2\rangle 0a, SafeAcceptorIsAcceptor$ DEF $TypeOK$
      $\langle 6\rangle 7.$CASE $p \in 2avSent[acc]$BY $\langle 6\rangle 7$
      $\langle 6\rangle 8.$CASE $p \notin 2avSent[acc]$
        $\langle 7\rangle 1.$ $p = [lr \mapsto lrn, bal \mapsto bal, val \mapsto val]$BY $\langle 6\rangle 8, \langle 2\rangle 0a, SafeAcceptorIsAcceptor$ DEF $TypeOK$
        $\langle 7\rangle 2.$ $maxBal[m.lr, m.acc] \leq bal$BY $\langle 5\rangle 2, \langle 7\rangle 1, \langle 6\rangle 6$
        $\langle 7\rangle 4.$ $m.bal \leq bal$BY $\langle 6\rangle 1, \langle 7\rangle 2, \langle 6\rangle 3, \langle 2\rangle 0g, BallotLeqTrans$
        $\langle 7\rangle 10.$ QED BY $\langle 7\rangle 1, \langle 7\rangle 4, \langle 6\rangle 6, \langle 6\rangle 3, BallotLeNotLeq$
      $\langle 6\rangle 10.$ QED BY $\langle 6\rangle 7, \langle 6\rangle 8$
    $\langle 5\rangle 3.$ QED BY $\langle 5\rangle 1, \langle 5\rangle 2$
  $\langle 4\rangle 10.$ QED BY $\langle 4\rangle 2, \langle 4\rangle 4, \langle 4\rangle 5$  DEF $MsgInv1b$
$\langle 3\rangle 3.$CASE $Phase2b(lrn, bal, acc, val)$
  $\langle 4\rangle$ SUFFICES
        ASSUME $\forall L \in Learner : maxBal[L, acc] \leq bal,$
               $Send([type \mapsto \text{“2b”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]),$
               $votesSent' = [votesSent$ EXCEPT
                          $![acc] = votesSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
        PROVE $MsgInv1b(m)'$
        BY $\langle 3\rangle 3$  DEF $Phase2b$
  $\langle 4\rangle 1.$ $m \in msgs$BY $\langle 2\rangle 0e$  DEF $Send$
  $\langle 4\rangle 1a.$ $m.acc \in Acceptor$BY $\langle 4\rangle 1, MessageType, \langle 2\rangle 0e, \langle 2\rangle 0$  DEF $TypeOK$
```

```
    ⟨4⟩2. $(m.bal \leq maxBal[m.lr, m.acc])'$BY ⟨4⟩1, ⟨1⟩1b, ⟨3⟩3  DEF $Phase2b$, $MsgInv1b$
    ⟨4⟩4. $(m.proposals = \{p \in 2avSent[m.acc] : p.bal < m.bal \land p.lr = m.lr\})'$
          BY ⟨4⟩1, ⟨1⟩1b, ⟨3⟩3  DEF $Phase2b$, $MsgInv1b$
    ⟨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 $MsgInv1b$
      ⟨5⟩2.CASE $m.acc = acc$
        ⟨6⟩1. $m.bal \leq maxBal[m.lr, m.acc]$BY ⟨1⟩1b, ⟨4⟩1  DEF $MsgInv1b$
        ⟨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 $MsgInv1b$
        ⟨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 $MsgInv1b$
  ⟨3⟩4. QED BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3
⟨2⟩4.CASE $AcceptorReceiveAction$BY ⟨1⟩1b, ⟨2⟩4  DEF $AcceptorReceiveAction$, $Recv$, $MsgInv1b$, $Next$
⟨2⟩5.CASE $AcceptorDisconnectAction$BY ⟨1⟩1b, ⟨2⟩5  DEF $AcceptorDisconnectAction$, $Disconnect$, $MsgInv1b$,
⟨2⟩6.CASE $LearnerAction$BY ⟨1⟩1b, ⟨2⟩6  DEF $LearnerAction$, $LearnerRecv$, $LearnerDecide$, $MsgInv1b$, $Next$
⟨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 $TypeOK$, $Next$,
               $\forall m \in msgs : m.acc \in SafeAcceptor \land m.type = \text{“2av”} \Rightarrow MsgInv2av(m)$,
               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 $ProposerAction$, $Phase1a$, $Phase1c$, $MsgInv2av$, $Next$, $Send$
  ⟨3⟩1. QED BY ⟨1⟩2av, ⟨3⟩0, ⟨2⟩1, ⟨2⟩0e  DEF ProposerAction, Phase1a, Phase1c, MsgInv2av, Next, Send
⟨2⟩2.CASE AcceptorSendAction
  ⟨3⟩ SUFFICES ASSUME NEW lrn ∈ Learner,
                      NEW bal ∈ Ballot,
                      NEW acc ∈ SafeAcceptor,
                      NEW val ∈ Value,
                      ∨ Phase1b(lrn, bal, acc)
                      ∨ Phase2av(lrn, bal, acc, val)
                      ∨ Phase2b(lrn, bal, acc, val)
               PROVE  MsgInv2av(m)'
    BY ⟨2⟩2  DEF AcceptorSendAction
  ⟨3⟩1.CASE Phase1b(lrn, bal, acc)
    ⟨4⟩1. m ∈ msgsBY ⟨3⟩1, ⟨2⟩0e  DEF Phase1b, Send
    ⟨4⟩2. QED BY ⟨1⟩2av, ⟨4⟩1, ⟨3⟩1  DEF Phase1b, MsgInv2av, 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 ↦ "2av", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val]),
                  2avSent' = [2avSent EXCEPT ![acc] =
                                2avSent[acc] ∪ {[lr ↦ lrn, bal ↦ bal, val ↦ val]}],
                  UNCHANGED received
              PROVE  MsgInv2av(m)'
          BY ⟨3⟩2  DEF Phase2av
    ⟨4⟩1.CASE m ∈ msgs
      ⟨5⟩1. InitializedBallot(m.lr, m.bal)'BY ⟨4⟩1, ⟨2⟩0e, ⟨1⟩2av, MsgsMonotone DEF MsgInv2av, Initialized
      ⟨5⟩2. AnnouncedValue(m.lr, m.bal, m.val)'BY ⟨4⟩1, ⟨2⟩0e, ⟨1⟩2av, MsgsMonotone DEF MsgInv2av, An
      ⟨5⟩3. KnowsSafeAt(m.lr, m.acc, m.bal, m.val)'BY ⟨4⟩1, ⟨1⟩2av  DEF Phase2av, MsgInv2av
      ⟨5⟩4. [lr ↦ m.lr, bal ↦ m.bal, val ↦ m.val] ∈ 2avSent'[m.acc]
          BY ⟨4⟩1, ⟨2⟩0e, ⟨1⟩2av, 2avSentMonotone, MessageType DEF MsgInv2av, TypeOK
      ⟨5⟩5. (∃ Q ∈ ByzQuorum :
              ∧ [lr ↦ m.lr, q ↦ Q] ∈ TrustLive
              ∧ ∀ ba ∈ Q :
                  ∃ m1b ∈ received[m.acc] :
                    ∧ m1b.type = "1b"
                    ∧ m1b.lr = m.lr
                    ∧ m1b.acc = ba
                    ∧ m1b.bal = m.bal)'
            BY ⟨4⟩1, ⟨2⟩0e, ⟨1⟩2av, 2avSentMonotone, MessageType DEF MsgInv2av, TypeOK
      ⟨5⟩6. QED BY ⟨5⟩1, ⟨5⟩2, ⟨5⟩3, ⟨5⟩4, ⟨5⟩5  DEF MsgInv2av
    ⟨4⟩2.CASE m ∉ msgs
      ⟨5⟩1. m = [type ↦ "2av", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val]BY ⟨4⟩2  DEF Send
      ⟨5⟩3. InitializedBallot(m.lr, m.bal)'BY ⟨5⟩1, ⟨3⟩2  DEF Phase2av
      ⟨5⟩4. AnnouncedValue(m.lr, m.bal, m.val)'BY ⟨5⟩1
      $\langle 5\rangle 5.$ $KnowsSafeAt(m.lr, m.acc, m.bal, m.val)'$BY $\langle 5\rangle 1$
      $\langle 5\rangle 6.$ $([lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in 2avSent[m.acc])'$BY $\langle 5\rangle 1$, $\langle 2\rangle 0$b  DEF $TypeOK$
      $\langle 5\rangle 7.$ $(\exists Q \in ByzQuorum :$
              $\land [lr \mapsto m.lr, q \mapsto Q] \in TrustLive$
              $\land \forall ba \in Q :$
                  $\exists m1b \in received[m.acc] :$
                     $\land m1b.type =$ "1b"
                     $\land m1b.lr = m.lr$
                     $\land m1b.acc = ba$
                     $\land m1b.bal = m.bal)'$
        $\langle 6\rangle 1.$CASE $KnowsSafeAt1(lrn, acc, bal, val)$
          $\langle 7\rangle 1.$ PICK $Q1 \in ByzQuorum :$
                         $\land [lr \mapsto lrn, q \mapsto Q1] \in TrustLive$
                         $\land \forall a \in Q1 :$
                             $\exists m1b \in received[acc] :$
                                 $\land$  $m1b.type =$ "1b"
                                 $\land$  $m1b.lr = lrn$
                                 $\land$  $m1b.bal = bal$
                                 $\land$  $m1b.acc = a$
                BY $\langle 6\rangle 1$  DEF $KnowsSafeAt1$
          $\langle 7\rangle 2.$ WITNESS $Q1 \in ByzQuorum$
          $\langle 7\rangle 3.$ QED BY $\langle 7\rangle 1$, $\langle 5\rangle 1$
        $\langle 6\rangle 2.$CASE $KnowsSafeAt2(lrn, acc, bal, val)$
          $\langle 7\rangle 1.$ PICK $Q2 \in ByzQuorum :$
                         $\land [lr \mapsto lrn, q \mapsto Q2] \in TrustLive$
                         $\land \forall a \in Q2 :$
                             $\exists m1b \in received[acc] :$
                                 $\land$  $m1b.type =$ "1b"
                                 $\land$  $m1b.lr = lrn$
                                 $\land$  $m1b.bal = bal$
                                 $\land$  $m1b.acc = a$
                BY $\langle 6\rangle 2$  DEF $KnowsSafeAt2$
          $\langle 7\rangle 2.$ WITNESS $Q2 \in ByzQuorum$
          $\langle 7\rangle 3.$ QED BY $\langle 7\rangle 1$, $\langle 5\rangle 1$
        $\langle 6\rangle 3.$ QED BY $\langle 6\rangle 1$, $\langle 6\rangle 2$  DEF $KnowsSafeAt$
      $\langle 5\rangle 8.$ QED BY $\langle 5\rangle 1$, $\langle 5\rangle 3$, $\langle 5\rangle 4$, $\langle 5\rangle 5$, $\langle 5\rangle 6$, $\langle 5\rangle 7$, $MessageType$ DEF $MsgInv2av$, $TypeOK$
    $\langle 4\rangle 20.$ QED BY $\langle 4\rangle 1$, $\langle 4\rangle 2$
  $\langle 3\rangle 3.$CASE $Phase2b(lrn, bal, acc, val)$
    $\langle 4\rangle 1.$ $m \in msgs$BY $\langle 3\rangle 3$, $\langle 2\rangle 0$e  DEF $Phase2b$, $Send$
    $\langle 4\rangle 2.$ QED BY $\langle 1\rangle 2$av, $\langle 4\rangle 1$, $\langle 3\rangle 3$  DEF $Phase2b$, $MsgInv2av$, $Send$
  $\langle 3\rangle 4.$ QED BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$
$\langle 2\rangle 4.$CASE $AcceptorReceiveAction$
  $\langle 3\rangle 1.$ $m \in msgs$BY $\langle 2\rangle 4$  DEF $AcceptorReceiveAction$, $Recv$
  $\langle 3\rangle 6.$ $(\exists Q \in ByzQuorum :$
          $\land [lr \mapsto m.lr, q \mapsto Q] \in TrustLive$
```

```
              $\land \forall ba \in Q :$
                  $\exists m1b \in received[m.acc] :$
                    $\land m1b.type = \text{"1b"}$
                    $\land m1b.lr = m.lr$
                    $\land m1b.acc = ba$
                    $\land m1b.bal = m.bal)'$
      ⟨7⟩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 ⟨1⟩2av, ⟨3⟩1, ⟨2⟩0e DEF $MsgInv2av$
      ⟨7⟩2. WITNESS $Q0 \in ByzQuorum$
      ⟨7⟩3. QED BY ⟨1⟩2av, ⟨7⟩1, $ReceivedMonotone$, $MessageType$, ⟨3⟩1 DEF $MsgInv2av$, $TypeOK$
    ⟨3⟩20. QED BY ⟨1⟩2av, ⟨2⟩4, ⟨3⟩6, $MessageType$, $ReceivedMonotone$ DEF $MsgInv2av$, $AcceptorReceiveActio$
  ⟨2⟩5.CASE $AcceptorDisconnectAction$BY ⟨1⟩2av, ⟨2⟩5 DEF $AcceptorDisconnectAction$, $Disconnect$, $MsgInv2a$
  ⟨2⟩6.CASE $LearnerAction$BY ⟨1⟩2av, ⟨2⟩6 DEF $LearnerAction$, $LearnerRecv$, $LearnerDecide$, $MsgInv2av$, $Ne$
  ⟨2⟩7.CASE $FakeAcceptorAction$
          BY ⟨1⟩2av, ⟨2⟩7, $SafeAcceptorAssumption$ DEF $FakeAcceptorAction$, $FakeSend$, $MsgInv2av$, $Send$
  ⟨2⟩8. QED BY ⟨1⟩2av, ⟨2⟩0b, ⟨2⟩1, ⟨2⟩2, ⟨2⟩4, ⟨2⟩5, ⟨2⟩6, ⟨2⟩7 DEF $Next$
⟨1⟩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 $MsgInv2b(m)'$
  ⟨2⟩0a. $TypeOK$BY ⟨1⟩2b
  ⟨2⟩0b. $TypeOK'$BY ⟨1⟩2b, $TypeOKInvariant$
  ⟨2⟩0c. $m \in Message$BY ⟨2⟩0b DEF $TypeOK$
  ⟨2⟩0d. $m.acc \in SafeAcceptor$BY ⟨1⟩2b
  ⟨2⟩0e. $m.type = \text{"2b"}$BY ⟨1⟩2b
  ⟨2⟩1.CASE $ProposerAction$
    ⟨3⟩1. $m \in msgs$BY ⟨2⟩1, ⟨2⟩0e DEF $ProposerAction$, $Phase1a$, $Phase1c$, $Send$
    ⟨3⟩10. QED BY ⟨1⟩2b, ⟨2⟩1, ⟨2⟩0a, ⟨2⟩0b, ⟨2⟩0d, ⟨2⟩0e, ⟨3⟩1
              DEF $TypeOK$, $ProposerAction$, $Phase1a$, $Phase1c$, $MsgInv2b$, $Next$, $Send$
  ⟨2⟩2.CASE $AcceptorSendAction$
    ⟨3⟩ HIDE DEF $Next$
    ⟨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 $MsgInv2b(m)'$
    BY $\langle 2\rangle 2$  DEF $AcceptorSendAction$
$\langle 3\rangle 1.$CASE $Phase1b(lrn, bal, acc)$
  $\langle 4\rangle 1.$ $m \in msgs$BY $\langle 3\rangle 1$, $\langle 2\rangle 0a$, $\langle 2\rangle 0e$  DEF $Phase1b$, $Send$, $TypeOK$
  $\langle 4\rangle 2.$ QED BY $\langle 1\rangle 2b$, $\langle 3\rangle 1$, $\langle 2\rangle 0a$, $\langle 2\rangle 0b$, $\langle 2\rangle 0e$, $\langle 4\rangle 1$  DEF $Phase1b$, $MsgInv2b$, $Send$, $TypeOK$
$\langle 3\rangle 2.$CASE $Phase2av(lrn, bal, acc, val)$
  $\langle 4\rangle 1.$ $m \in msgs$BY $\langle 3\rangle 2$, $\langle 2\rangle 0a$, $\langle 2\rangle 0e$  DEF $Phase2av$, $Send$, $TypeOK$
  $\langle 4\rangle 2.$ QED BY $\langle 1\rangle 2b$, $\langle 3\rangle 2$, $\langle 2\rangle 0a$, $\langle 2\rangle 0d$, $\langle 2\rangle 0e$, $\langle 4\rangle 1$  DEF $Phase2av$, $MsgInv2b$, $Send$, $TypeOK$
$\langle 3\rangle 3.$CASE $Phase2b(lrn, bal, acc, val)$
  $\langle 4\rangle 1.$CASE $m \in msgs$
    $\langle 5\rangle 1.$ $([lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in votesSent[m.acc])'$
      BY $\langle 1\rangle 2b$, $\langle 2\rangle 0e$, $\langle 4\rangle 1$, $MessageType$, $VotesSentMonotone$ DEF $MsgInv2b$, $TypeOK$
    $\langle 5\rangle 2.$ $(\exists Q \in ByzQuorum :$
            $\land [lr \mapsto m.lr, q \mapsto Q] \in TrustLive$
            $\land \forall ba \in Q :$
                $\exists m2av \in received[m.acc] :$
                    $\land m2av.type = \text{“2av”}$
                    $\land m2av.lr = m.lr$
                    $\land m2av.acc = ba$
                    $\land m2av.bal = m.bal$
                    $\land m2av.val = m.val)'$
        BY $\langle 1\rangle 2b$, $\langle 3\rangle 3$, $\langle 2\rangle 0a$, $\langle 2\rangle 0b$, $\langle 2\rangle 0d$, $\langle 2\rangle 0e$, $\langle 4\rangle 1$  DEF $Phase2b$, $MsgInv2b$, $Send$, $TypeOK$
    $\langle 5\rangle 3.$ QED BY $\langle 5\rangle 1$, $\langle 5\rangle 2$  DEF $MsgInv2b$
  $\langle 4\rangle 2.$CASE $m \notin msgs$
    $\langle 5\rangle$ SUFFICES
        ASSUME NEW $Q \in ByzQuorum$,
                $[lr \mapsto lrn, q \mapsto Q] \in TrustLive$,
                $\forall aa \in Q :$
                    $\exists m\_1 \in \{mm \in received[acc] :$
                                $\land mm.type = \text{“2av”}$
                                $\land mm.lr = lrn$
                                $\land mm.bal = bal\} :$
                        $\land m\_1.val = val$
                        $\land m\_1.acc = aa$,
                $Send([type \mapsto \text{“2b”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val])$,
                $votesSent' = [votesSent$ EXCEPT $![acc] =$
                                $votesSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
                PROVE $MsgInv2b(m)'$
        BY $\langle 3\rangle 3$  DEF $Phase2b$
    $\langle 5\rangle 1.$ $m = [type \mapsto \text{“2b”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]$BY $\langle 4\rangle 2$  DEF $Send$
    $\langle 5\rangle 1e.$ UNCHANGED $received$BY $\langle 3\rangle 3$  DEF $Phase2b$
    $\langle 5\rangle 2.$ $([lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in votesSent[m.acc])'$
      BY $\langle 5\rangle 1$, $\langle 2\rangle 0a$, $\langle 2\rangle 0b$, $\langle 2\rangle 0e$, $MessageType$ DEF $TypeOK$
    $\langle 5\rangle 3.$ $(\exists Q\_1 \in ByzQuorum :$
          $\land [lr \mapsto m.lr, q \mapsto Q\_1] \in TrustLive$
          $\land \forall ba \in Q\_1 :$
```

```
                $\exists\, m2av \in received[m.acc] :$
                  $\land\ m2av.type = $ “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 = $ “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 = $ “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 $TypeOK$
    $\langle 3\rangle 8.$ QED BY $\langle 3\rangle 3, \langle 3\rangle 7$ DEF $MsgInv2b$
  $\langle 2\rangle 5.$CASE $AcceptorDisconnectAction$BY $\langle 1\rangle 2b, \langle 2\rangle 5$ DEF $AcceptorDisconnectAction, Disconnect, MsgInv2b,$
  $\langle 2\rangle 6.$CASE $LearnerAction$BY $\langle 1\rangle 2b, \langle 2\rangle 6$ DEF $LearnerAction, LearnerRecv, LearnerDecide, MsgInv2b, Next$
  $\langle 2\rangle 7.$CASE $FakeAcceptorAction$
      BY $\langle 1\rangle 2b, \langle 2\rangle 7, SafeAcceptorAssumption$ DEF $FakeAcceptorAction, FakeSend, MsgInv2b, Send$
  $\langle 2\rangle 8.$ QED BY $\langle 1\rangle 2b, \langle 2\rangle 0a, \langle 2\rangle 1, \langle 2\rangle 2, \langle 2\rangle 4, \langle 2\rangle 5, \langle 2\rangle 6, \langle 2\rangle 7$ DEF $Next$
$\langle 1\rangle 3.$ QED BY $\langle 1\rangle 1b, \langle 1\rangle 2av, \langle 1\rangle 2b$

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

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

LEMMA $HeterogeneousSpecInvariant \triangleq$
    $TypeOK \land Next \land ReceivedSpec \land$
    $2avSentSpec1 \land$
    $VotesSentSpec2 \land VotesSentSpec3 \land VotesSentSpec4 \land$
    $ConnectedSpec \land MsgInv \land$
    $HeterogeneousSpec \Rightarrow HeterogeneousSpec'$
PROOF
$\langle 1\rangle$ SUFFICES ASSUME $TypeOK, Next, ReceivedSpec, 2avSentSpec1, VotesSentSpec2, VotesSentSpec3, VotesSen$
                        $ConnectedSpec, MsgInv, HeterogeneousSpec,$
                        NEW $L1 \in Learner$, NEW $L2 \in Learner$,
                        NEW $B1 \in Ballot$, NEW $B2 \in Ballot$,
                        NEW $V1 \in Value$, NEW $V2 \in Value$,
                        NEW $A2 \in SafeAcceptor$,
                        NEW $Q1 \in ByzQuorum$,
                        NEW $m \in msgs'$,
                        $\langle L1, L2\rangle \in Ent$,
                        $[lr \mapsto L1, q \mapsto Q1] \in TrustLive$,
                    $m.type = \text{"2av"}, m.lr = L2, m.acc = A2, m.bal = B2, m.val = V2,$
                    $B1 < B2,$
                    $V1 \neq V2$
               PROVE $CannotDecide(Q1, L1, B1, V1)'$
    BY DEF $HeterogeneousSpec$
$\langle 1 \rangle$ USE DEF $MsgInv$
$\langle 1 \rangle$0a. $TypeOK$OBVIOUS
$\langle 1 \rangle$0b. $TypeOK'$BY $TypeOKInvariant$
$\langle 1 \rangle$0c. $m \in Message$BY $\langle 1 \rangle$0b DEF $TypeOK$
$\langle 1 \rangle$1.CASE $ProposerAction$BY $\langle 1 \rangle$1 DEF $ProposerAction, Phase1a, Phase1c, Next, Send, HeterogeneousSpec$
$\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 $CannotDecide(Q1, L1, B1, V1)'$
    BY $\langle 1 \rangle$2 DEF $AcceptorSendAction$
  $\langle 2 \rangle$1.CASE $Phase1b(lrn, bal, acc)$
    $\langle 3 \rangle$1. $m \in msgs$BY $\langle 2 \rangle$1, $\langle 1 \rangle$0b DEF $Phase1b, Send, TypeOK, Message$
    $\langle 3 \rangle$2. QED BY $\langle 3 \rangle$1 DEF $HeterogeneousSpec$
  $\langle 2 \rangle$2.CASE $Phase2av(lrn, bal, acc, val)$
    $\langle 3 \rangle$0. $msgs \subseteq msgs'$BY $\langle 2 \rangle$2 DEF $Phase2av, Send$
    $\langle 3 \rangle$1.CASE $m \in msgs$BY $\langle 3 \rangle$1 DEF $HeterogeneousSpec$
    $\langle 3 \rangle$2.CASE $m \notin msgs$
      $\langle 4 \rangle$0. $m = [type \mapsto \text{"2av"}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val]$
            BY $\langle 3 \rangle$2, $\langle 2 \rangle$2 DEF $Phase2av, Send$
      $\langle 4 \rangle$0a. $lrn = L2 \land acc = A2 \land bal = B2 \land val = V2$BY $\langle 4 \rangle$0
      $\langle 4 \rangle$1. $maxBal[L2, A2] \leq B2$BY $\langle 2 \rangle$2, $\langle 4 \rangle$0a DEF $Phase2av$
      $\langle 4 \rangle$2. $KnowsSafeAt(L2, A2, B2, V2)$BY $\langle 2 \rangle$2, $\langle 4 \rangle$0a DEF $Phase2av$
      $\langle 4 \rangle$3a.CASE $KnowsSafeAt1(L2, A2, B2, V2)$
        $\langle 5 \rangle$0. USE DEF $CannotDecide$
        $\langle 5 \rangle$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 $\langle 4 \rangle$3a DEF $KnowsSafeAt1$
    $\langle 5\rangle 2.$ PICK $S \in SafeAcceptor : S \in Q1 \land S \in Q2$BY $EntanglementTrustLive$, $\langle 4\rangle 0$, $\langle 5\rangle 1$
    $\langle 5\rangle 3.$ PICK $m1b \in received[A2]$ :
                  $\land$ $m1b.type =$ "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 $\langle 5\rangle 1$, $\langle 5\rangle 2$
    $\langle 5\rangle 4.$ $\land\, m1b \in msgs$
         $\land\, m1b.type =$ "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 $\langle 5\rangle 3$, $SafeAcceptorIsAcceptor$ DEF $TypeOK$, $ReceivedSpec$
    $\langle 5\rangle 5.$ WITNESS $S \in SafeAcceptor$
    $\langle 5\rangle 6.$ $\exists L \in Learner : LeftBallot(L, S, B1)'$BY $\langle 5\rangle 4$, $\langle 3\rangle 0$  DEF $LeftBallot$
    $\langle 5\rangle 7.$ $\neg VotedFor(L1, S, B1, V1)'$
      $\langle 6\rangle 1.$ SUFFICES ASSUME $VotedFor(L1, S, B1, V1)$PROVE FALSEOBVIOUS
      $\langle 6\rangle 2.$ $[lr \mapsto L1, bal \mapsto B1, val \mapsto V1] \in votesSent[S]$BY $\langle 6\rangle 1$  DEF $VotesSentSpec2$
      $\langle 6\rangle 3.$ $m1b.votes = \{p \in votesSent[S] : MaxVote(S, B2, p)\}$BY $\langle 5\rangle 4$  DEF $MsgInv1b$
      $\langle 6\rangle 4.$ PICK $P \in votesSent[S] : MaxVote(S, B2, P) \land P.lr = L1 \land B1 \leq P.bal$
        $\langle 7\rangle 1.$ 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 $\langle 6\rangle 2$
        $\langle 7\rangle 2.$ $P0.bal < B2$BY $\langle 7\rangle 1$
        $\langle 7\rangle 3.$ QED BY $\langle 7\rangle 1$, $\langle 7\rangle 2$  DEF $VotesSentSpec3$
      $\langle 6\rangle 5.$ $P \in m1b.votes$BY $\langle 6\rangle 3$, $\langle 6\rangle 4$
      $\langle 6\rangle 6.$ $\langle P.lr, L2\rangle \in connected[A2]$BY $\langle 6\rangle 4$  DEF $ConnectedSpec$
      $\langle 6\rangle 7.$ $B2 \leq P.bal$BY $\langle 6\rangle 5$, $\langle 6\rangle 6$, $\langle 5\rangle 4$
      $\langle 6\rangle 8.$ $P \in [lr : Learner, bal : Ballot, val : Value]$BY $\langle 6\rangle 4$, $SafeAcceptorIsAcceptor$ DEF $TypeOK$
      $\langle 6\rangle 9.$ $P.bal \in Ballot$BY $\langle 6\rangle 8$
      $\langle 6\rangle 10.$ QED BY $\langle 6\rangle 9$, $\langle 6\rangle 7$, $\langle 6\rangle 4$, $BallotLeNotLeq$ DEF $MaxVote$
    $\langle 5\rangle 8.$ QED BY $\langle 5\rangle 2$, $\langle 5\rangle 6$, $\langle 5\rangle 7$
  $\langle 4\rangle 3$b.CASE $KnowsSafeAt2(L2, A2, B2, V2)$
    $\langle 5\rangle 1.$ PICK $c \in Ballot, BQ \in ByzQuorum, WQ \in ByzQuorum$ :
                  $\land\, c < B2$
                  $\land\, [lr \mapsto L2, q \mapsto BQ] \in TrustLive$
                  $\land\, \forall a \in BQ$ :
                      $\exists m1 \in \{mm \in received[A2] : mm.type =$ "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 $KnowsSafeAt2$, $Ballot$
$\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 $VotedFor$, $Phase2av$, $Send$
  $\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 $VotesSentSpec2$
  $\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 $VotesSentSpec3$, $TypeOK$
  $\langle 6\rangle$3. $P \in m1.votes$BY $\langle 6\rangle$2, $\langle 5\rangle$5  DEF $MsgInv1b$
  $\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 $P.val = V1$
```

```
      $\langle 8\rangle 1.$ $P.bal \leq c \land (P.bal = c \Rightarrow P.val = V2)$BY $\langle 5\rangle 5$, $\langle 6\rangle 3$, $\langle 6\rangle 4$
      $\langle 8\rangle 2.$ $P.bal < c$BY $\langle 6\rangle 5$, $\langle 8\rangle 1$, $\langle 7\rangle 1$  DEF $Ballot$
      $\langle 8\rangle 10.$ QED BY $\langle 6\rangle 2$, $\langle 6\rangle 5$, $\langle 8\rangle 2$, $BallotLeqLeTrans$
    $\langle 7\rangle 2.$CASE $P.val \neq V1$
      $\langle 8\rangle 1.$ $B1 < P.bal$
        $\langle 9\rangle 0.$ $\langle L1, L1\rangle \in Ent$BY $EntanglementSelf$
        $\langle 9\rangle 1.$ $B1 \leq P.bal$BY $\langle 6\rangle 2$
        $\langle 9\rangle 2.$ $B1 \neq P.bal$BY $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 6\rangle 5$, $\langle 7\rangle 2$, $\langle 9\rangle 0$  DEF $VotesSentSpec4$
        $\langle 9\rangle 3.$ QED BY $\langle 6\rangle 5$, $\langle 9\rangle 1$, $\langle 9\rangle 2$  DEF $Ballot$
      $\langle 8\rangle 2.$ $P.bal \leq c$BY $\langle 5\rangle 5$, $\langle 6\rangle 3$, $\langle 6\rangle 4$
      $\langle 8\rangle 3.$ QED BY $\langle 8\rangle 1$, $\langle 8\rangle 2$, $\langle 6\rangle 5$, $BallotLeLeqTrans$
    $\langle 7\rangle 3.$ QED BY $\langle 7\rangle 1$, $\langle 7\rangle 2$
  $\langle 6\rangle 7.$ PICK $S2 \in SafeAcceptor : S2 \in Q1 \land S2 \in WQ$BY $EntanglementTrustLive$, $\langle 4\rangle 0$, $\langle 5\rangle 1$
  $\langle 6\rangle 8.$ 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 $\langle 5\rangle 1$, $\langle 6\rangle 7$
  $\langle 6\rangle 9.$ 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 $\langle 6\rangle 8$, $SafeAcceptorIsAcceptor$ DEF $TypeOK$, $ReceivedSpec$
  $\langle 6\rangle 10.$ $Proposed(L2, S2, c, V2)$
    $\langle 7\rangle 1.$ $p2 \in 2avSent[S2]$BY $\langle 6\rangle 9$  DEF $MsgInv1b$
    $\langle 7\rangle 2.$ QED BY $\langle 7\rangle 1$, $\langle 6\rangle 9$  DEF $2avSentSpec1$
  $\langle 6\rangle 11.$ 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 $\langle 6\rangle 10$  DEF $Proposed$
  $\langle 6\rangle 12.$ SUFFICES $CannotDecide(Q1, L1, B1, V1)$BY  DEF $CannotDecide$
  $\langle 6\rangle 15.$ QED BY $\langle 6\rangle 11$, $\langle 6\rangle 6$  DEF $HeterogeneousSpec$
 $\langle 5\rangle 8.$ QED BY $\langle 5\rangle 6$, $\langle 5\rangle 7$
$\langle 4\rangle 4.$ QED BY $\langle 4\rangle 3a$, $\langle 4\rangle 3b$, $\langle 4\rangle 2$  DEF $KnowsSafeAt$
$\langle 3\rangle 3.$ QED BY $\langle 3\rangle 1$, $\langle 3\rangle 2$
```

```
  ⟨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 $Chosen$
```

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

LEMMA $\mathit{ChosenSafeCaseLt}\ \triangleq$
   ASSUME 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 $V2 \in \mathit{Value}$,
              $\mathit{TypeOK}$, $\mathit{ReceivedSpec}$, $\mathit{ReceivedByLearnerSpec}$, $\mathit{MsgInv}$,
              $\mathit{HeterogeneousSpec}$,
              $\langle L1,\ L2 \rangle \in \mathit{Ent}$,
              $B1 < B2$,
              $\mathit{ChosenIn}(L1,\ B1,\ V1)$, $\mathit{ChosenIn}(L2,\ B2,\ V2)$
   PROVE $V1 = V2$
PROOF
$\langle 1 \rangle$ USE  DEF $\mathit{MsgInv}$
$\langle 1 \rangle$ SUFFICES ASSUME $V1 \neq V2$PROVE FALSEOBVIOUS
$\langle 1 \rangle 1.$ PICK $Q1 \in \mathit{ByzQuorum}$ :
        $\land$  $[\mathit{lr} \mapsto L1,\ q \mapsto Q1] \in \mathit{TrustLive}$
        $\land$  $\forall\, aa\ \in Q1$ :
              $\exists\, m \in \{mm \in \mathit{receivedByLearner}[L1] : mm.\mathit{bal} = B1\}$ :
                  $\land\ m.\mathit{val}\ = V1$
                  $\land\ m.\mathit{acc}\ = aa$
     BY  DEF $\mathit{ChosenIn}$
$\langle 1 \rangle 2.$ PICK $Q2 \in \mathit{ByzQuorum}$ :
        $\land$  $[\mathit{lr} \mapsto L2,\ q \mapsto Q2] \in \mathit{TrustLive}$
        $\land$  $\forall\, aa\ \in Q2$ :
              $\exists\, m \in \{mm \in \mathit{receivedByLearner}[L2] : mm.\mathit{bal} = B2\}$ :
                  $\land\ m.\mathit{val}\ = V2$
                  $\land\ m.\mathit{acc}\ = aa$
      BY  DEF $ChosenIn$
$\langle 1\rangle 3.$ PICK $A \in SafeAcceptor : A \in Q1 \land A \in Q2$BY $EntanglementTrustLive$, $\langle 1\rangle 1$, $\langle 1\rangle 2$
$\langle 1\rangle 4.$ PICK $m1 \in receivedByLearner[L1] : m1.acc = A \land m1.bal = B1 \land m1.val = V1$BY $\langle 1\rangle 1$, $\langle 1\rangle 3$  DEF $Chose$
$\langle 1\rangle 5.$ PICK $m2 \in receivedByLearner[L2] : m2.acc = A \land m2.bal = B2 \land m2.val = V2$BY $\langle 1\rangle 2$, $\langle 1\rangle 3$  DEF $Chose$
$\langle 1\rangle 6.$ $\land\ m1 \in msgs$
      $\land\ m1.type = \text{“2b”}$
      $\land\ m1.lr = L1$
      $\land\ m1.acc = A$
      $\land\ m1.bal = B1$
      $\land\ m1.val = V1$
     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 $ReceivedByLearnerSpec$, $TypeOK$
$\langle 1\rangle 10.$ PICK $R1 \in ByzQuorum$ :
              $\land\ [lr \mapsto L1,\ q \mapsto R1] \in TrustLive$
       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 $EntanglementTrustLive$, $\langle 1\rangle 10$, $\langle 1\rangle 11$
$\langle 1\rangle 14.$ 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 $\langle 1\rangle 12$, $\langle 1\rangle 11$
$\langle 1\rangle 16.$ $\land\ m2av2 \in msgs$
       $\land\ m2av2.type = \text{“2av”}$
       $\land\ m2av2.lr = L2$
       $\land\ m2av2.acc = A0$
       $\land\ m2av2.bal = B2$
       $\land\ m2av2.val = V2$
      BY $\langle 1\rangle 14$, $SafeAcceptorIsAcceptor$ DEF $ReceivedSpec$, $TypeOK$
$\langle 1\rangle 17.$ $CannotDecide(Q1,\ L1,\ B1,\ V1)$
  $\langle 2\rangle 1.$ $[lr \mapsto L1,\ q \mapsto Q1] \in TrustLive$BY $\langle 1\rangle 1$
  $\langle 2\rangle 5.$ QED BY $\langle 1\rangle 16$, $\langle 2\rangle 1$  DEF $HeterogeneousSpec$
```

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

LEMMA $\mathit{ChosenSafe} \triangleq$
    ASSUME 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 $V2 \in \mathit{Value}$,
                  $\mathit{TypeOK}, \mathit{ReceivedSpec}, \mathit{ReceivedByLearnerSpec}, \mathit{VotesSentSpec4}, \mathit{MsgInv},$
                  $\mathit{HeterogeneousSpec},$
                  $\langle L1, L2\rangle \in \mathit{Ent},$
                  $\mathit{ChosenIn}(L1, B1, V1), \mathit{ChosenIn}(L2, B2, V2)$
    PROVE $V1 = V2$
PROOF
$\langle 1\rangle$ USE  DEF $\mathit{MsgInv}$
$\langle 1\rangle 1.$ PICK $Q1 \in \mathit{ByzQuorum}$ :
          $\land$  $[\mathit{lr} \mapsto L1, q \mapsto Q1] \in \mathit{TrustLive}$
          $\land$  $\forall\, aa \in Q1 :$
                $\exists\, m \in \{mm \in \mathit{receivedByLearner}[L1] : mm.\mathit{bal} = B1\} :$
                    $\land\ m.\mathit{val} = V1$
                    $\land\ m.\mathit{acc} = aa$
       BY  DEF $\mathit{ChosenIn}$
$\langle 1\rangle 2.$ PICK $Q2 \in \mathit{ByzQuorum}$ :
          $\land$  $[\mathit{lr} \mapsto L2, q \mapsto Q2] \in \mathit{TrustLive}$
          $\land$  $\forall\, aa \in Q2 :$
                $\exists\, m \in \{mm \in \mathit{receivedByLearner}[L2] : mm.\mathit{bal} = B2\} :$
                    $\land\ m.\mathit{val} = V2$
                    $\land\ m.\mathit{acc} = aa$
       BY  DEF $\mathit{ChosenIn}$
$\langle 1\rangle 3.$ PICK $A \in \mathit{SafeAcceptor} : A \in Q1 \land A \in Q2$BY $\mathit{EntanglementTrustLive}, \langle 1\rangle 1, \langle 1\rangle 2$
$\langle 1\rangle 4.$ PICK $m1 \in \mathit{receivedByLearner}[L1] : m1.\mathit{acc} = A \land m1.\mathit{bal} = B1 \land m1.\mathit{val} = V1$BY $\langle 1\rangle 1, \langle 1\rangle 3$  DEF $\mathit{Chose}$
$\langle 1\rangle 5.$ PICK $m2 \in \mathit{receivedByLearner}[L2] : m2.\mathit{acc} = A \land m2.\mathit{bal} = B2 \land m2.\mathit{val} = V2$BY $\langle 1\rangle 2, \langle 1\rangle 3$  DEF $\mathit{Chose}$
$\langle 1\rangle 6.$ $\land\ m1 \in \mathit{msgs}$
      $\land\ m1.\mathit{type} = \text{“2b”}$
      $\land\ m1.\mathit{lr} = L1$
      $\land\ m1.\mathit{acc} = A$
      $\land\ m1.\mathit{bal} = B1$
      $\land\ m1.\mathit{val} = V1$
    BY $\langle 1\rangle 4$  DEF $\mathit{ReceivedByLearnerSpec}$, $\mathit{TypeOK}$
$\langle 1\rangle 7.$ $\land\ m2 \in msgs$
    $\land\ m2.type = $ “2b”
    $\land\ m2.lr = L2$
    $\land\ m2.acc = A$
    $\land\ m2.bal = B2$
    $\land\ m2.val = V2$
    BY $\langle 1\rangle 5$  DEF $\mathit{ReceivedByLearnerSpec}$, $\mathit{TypeOK}$
$\langle 1\rangle 8.$ $[lr \mapsto L1,\ bal \mapsto B1,\ val \mapsto V1] \in votesSent[A]$BY $\langle 1\rangle 6$  DEF $\mathit{MsgInv2b}$
$\langle 1\rangle 9.$ $[lr \mapsto L2,\ bal \mapsto B2,\ val \mapsto V2] \in votesSent[A]$BY $\langle 1\rangle 7$  DEF $\mathit{MsgInv2b}$
$\langle 1\rangle 10.$ 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 = $ “2av”
                    $\land\ m2av.lr = L1$
                    $\land\ m2av.acc = aa$
                    $\land\ m2av.bal = B1$
                    $\land\ m2av.val = V1$
      BY $\langle 1\rangle 6$  DEF $\mathit{MsgInv2b}$
$\langle 1\rangle 11.$ 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 = $ “2av”
                    $\land\ m2av.lr = L2$
                    $\land\ m2av.acc = aa$
                    $\land\ m2av.bal = B2$
                    $\land\ m2av.val = V2$
      BY $\langle 1\rangle 7$  DEF $\mathit{MsgInv2b}$
$\langle 1\rangle 12.$ PICK $A0 \in SafeAcceptor : A0 \in R1 \land A0 \in R2$BY $\mathit{EntanglementTrustLive}$, $\langle 1\rangle 10$, $\langle 1\rangle 11$
$\langle 1\rangle 13.$ PICK $m2av1 \in received[A]$ :
            $m2av1.type = $ “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 = $ “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 = $ “2av”
      $\land\ m2av1.lr = L1$
      $\land\ m2av1.acc = A0$
      $\land\ m2av1.bal = B1$
      $\land\ m2av1.val = V1$
     BY $\langle 1\rangle 13$, $\mathit{SafeAcceptorIsAcceptor}$ DEF $\mathit{ReceivedSpec}$, $\mathit{TypeOK}$
$\langle 1\rangle 16.$ $\land\ m2av2 \in msgs$
```

```
        ∧ m2av2.type = "2av"
        ∧ m2av2.lr = L2
        ∧ m2av2.acc = A0
        ∧ m2av2.bal = B2
        ∧ m2av2.val = V2
       BY ⟨1⟩14, SafeAcceptorIsAcceptor DEF ReceivedSpec, TypeOK
⟨1⟩30.CASE B1 < B2BY ⟨1⟩30, ChosenSafeCaseLt
⟨1⟩31.CASE B2 < B1BY ⟨1⟩31, ChosenSafeCaseLt, EntanglementSym
⟨1⟩32.CASE B1 = B2BY ⟨1⟩32, ChosenSafeCaseEq
⟨1⟩33. QED BY ⟨1⟩30, ⟨1⟩31, ⟨1⟩32, BallotOrderCases

Safety ≜
    ∀ L1, L2 ∈ Learner : ∀ B1, B2 ∈ Ballot : ∀ V1, V2 ∈ Value :
       ⟨L1, L2⟩ ∈ Ent ∧
       V1 ∈ decision[L1, B1] ∧ V2 ∈ decision[L2, B2] ⇒ V1 = V2

LEMMA SafetyStep ≜
    TypeOK ∧ Next ∧ MsgInv ∧
    DecisionSpec ∧ ReceivedSpec ∧ ReceivedByLearnerSpec ∧
    2avSentSpec1 ∧ 2avSentSpec3 ∧ VotesSentSpec4 ∧
    HeterogeneousSpec ∧ Safety ⇒ Safety′
PROOF
⟨1⟩ SUFFICES
        ASSUME TypeOK, Next, MsgInv, Safety, DecisionSpec, ReceivedSpec, ReceivedByLearnerSpec,
               2avSentSpec1, 2avSentSpec3, VotesSentSpec4,
               HeterogeneousSpec,
               NEW L1 ∈ Learner, NEW L2 ∈ Learner,
               NEW B1 ∈ Ballot, NEW B2 ∈ Ballot,
               NEW V1 ∈ Value, NEW V2 ∈ Value,
               ⟨L1, L2⟩ ∈ Ent,
               V1 ∈ decision′[L1, B1], V2 ∈ decision′[L2, B2]
        PROVE V1 = V2
    BY DEF Safety
⟨1⟩0a. TypeOKOBVIOUS
⟨1⟩0b. TypeOK′BY TypeOKInvariant
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1 DEF ProposerAction, Phase1a, Phase1c, Send, Safety
⟨1⟩2.CASE AcceptorSendAction
  ⟨2⟩ SUFFICES ASSUME NEW lrn ∈ Learner,
                      NEW bal ∈ Ballot,
                      NEW acc ∈ SafeAcceptor,
                      NEW val ∈ Value,
                      ∨ Phase1b(lrn, bal, acc)
                      ∨ Phase2av(lrn, bal, acc, val)
                      ∨ Phase2b(lrn, bal, acc, val)
               PROVE V1 = V2
```

```
      BY ⟨1⟩2  DEF AcceptorSendAction
  ⟨2⟩2.CASE Phase1b(lrn, bal, acc)BY ⟨2⟩2, ⟨1⟩0a, ⟨1⟩0b  DEF AcceptorSendAction, Send, Phase1b, Safety, Ty
  ⟨2⟩3.CASE Phase2av(lrn, bal, acc, val)BY ⟨2⟩3, ⟨1⟩0a, ⟨1⟩0b  DEF AcceptorSendAction, Send, Phase2av, Saf
  ⟨2⟩4.CASE Phase2b(lrn, bal, acc, val)BY ⟨2⟩4, ⟨1⟩0a, ⟨1⟩0b  DEF AcceptorSendAction, Send, Phase2b, Safet
  ⟨2⟩5. QED BY ⟨2⟩2, ⟨2⟩3, ⟨2⟩4
⟨1⟩3.CASE AcceptorReceiveActionBY ⟨1⟩3, ⟨1⟩0a, ⟨1⟩0b  DEF AcceptorReceiveAction, Recv, TypeOK, Safety
⟨1⟩4.CASE AcceptorDisconnectActionBY ⟨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 = V2BY ⟨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 ![⟨lrn, bal⟩] = decision[lrn, bal] ∪ {val}],
                        UNCHANGED ⟨msgs, maxBal, votesSent, 2avSent, received, connected, receivedByLe
                 PROVE  V1 = V2
      BY ⟨2⟩2  DEF LearnerDecide
    ⟨3⟩0.CASE V1 = V2BY ⟨3⟩0
    ⟨3⟩1.CASE V1 ≠ V2
      ⟨4⟩1.CASE val ≠ V1 ∧ val ≠ V2BY ⟨4⟩1  DEF Safety, TypeOK
      ⟨4⟩2.CASE val = V1
        ⟨5⟩0. V2 ∈ decision[L2, B2]BY ⟨3⟩1, ⟨4⟩2  DEF TypeOK
        ⟨5⟩1. ChosenIn(L2, B2, V2)BY ⟨5⟩0  DEF DecisionSpec
        ⟨5⟩2.CASE V1 ∈ decision[L1, B1]BY ⟨5⟩0, ⟨5⟩2  DEF Safety
        ⟨5⟩3.CASE V1 ∉ decision[L1, B1]
          ⟨6⟩1. lrn = L1 ∧ bal = B1BY ⟨5⟩3, ⟨4⟩2  DEF TypeOK
          ⟨6⟩2. ChosenIn(L1, B1, V1)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 ∈ decision[L1, B1]BY ⟨3⟩1, ⟨4⟩3  DEF TypeOK
        ⟨5⟩1. ChosenIn(L1, B1, V1)BY ⟨5⟩0  DEF DecisionSpec
        ⟨5⟩2.CASE V2 ∈ decision[L2, B2]BY ⟨5⟩0, ⟨5⟩2  DEF Safety
        ⟨5⟩3.CASE V2 ∉ decision[L2, B2]
          ⟨6⟩1. lrn = L2 ∧ bal = B2BY ⟨5⟩3, ⟨4⟩3  DEF TypeOK
          ⟨6⟩2. ChosenIn(L2, B2, V2)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 FakeAcceptorActionBY ⟨1⟩6  DEF FakeAcceptorAction, FakeSend, Send, Safety
```

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

$$\begin{array}{l}
\mathit{FullSafetyInvariant} \triangleq \\
\quad \land \mathit{TypeOK} \\
\quad \land \mathit{MsgInv} \\
\quad \land 2\mathit{avSentSpec}1 \land 2\mathit{avSentSpec}2 \land 2\mathit{avSentSpec}3 \\
\quad \land \mathit{VotesSentSpec}1 \land \mathit{VotesSentSpec}2 \land \mathit{VotesSentSpec}3 \land \mathit{VotesSentSpec}4 \\
\quad \land \mathit{ReceivedSpec} \\
\quad \land \mathit{ReceivedByLearnerSpec} \\
\quad \land \mathit{ConnectedSpec} \\
\quad \land \mathit{DecisionSpec} \\
\quad \land \mathit{HeterogeneousSpec} \\
\quad \land \mathit{Safety}
\end{array}$$

LEMMA $\mathit{TypeOKInit} \triangleq \mathit{Init} \Rightarrow \mathit{TypeOK}$
PROOF BY DEF Init, TypeOK

LEMMA $\mathit{MsgInvInit} \triangleq \mathit{Init} \Rightarrow \mathit{MsgInv}$
PROOF BY DEF Init, MsgInv

LEMMA $2\mathit{avSentSpec}1\mathit{Init} \triangleq \mathit{Init} \Rightarrow 2\mathit{avSentSpec}1$
PROOF BY DEF Init, $2\mathit{avSentSpec}1$

LEMMA $2\mathit{avSentSpec}2\mathit{Init} \triangleq \mathit{Init} \Rightarrow 2\mathit{avSentSpec}2$
PROOF BY DEF Init, $2\mathit{avSentSpec}2$, $\mathit{Proposed}$

LEMMA $2\mathit{avSentSpec}3\mathit{Init} \triangleq \mathit{Init} \Rightarrow 2\mathit{avSentSpec}3$
PROOF BY DEF Init, $2\mathit{avSentSpec}3$, TypeOK

LEMMA $\mathit{VotesSentSpec}1\mathit{Init} \triangleq \mathit{Init} \Rightarrow \mathit{VotesSentSpec}1$
PROOF BY DEF Init, $\mathit{VotesSentSpec}1$

LEMMA $\mathit{VotesSentSpec}2\mathit{Init} \triangleq \mathit{Init} \Rightarrow \mathit{VotesSentSpec}2$
PROOF BY DEF Init, $\mathit{VotesSentSpec}2$, $\mathit{VotedFor}$

LEMMA $\mathit{VotesSentSpec}3\mathit{Init} \triangleq \mathit{Init} \Rightarrow \mathit{VotesSentSpec}3$
PROOF BY DEF Init, $\mathit{VotesSentSpec}3$

LEMMA $\mathit{VotesSentSpec}4\mathit{Init} \triangleq \mathit{Init} \Rightarrow \mathit{VotesSentSpec}4$
PROOF BY DEF Init, $\mathit{VotesSentSpec}4$

LEMMA $\mathit{ReceivedSpecInit} \triangleq \mathit{Init} \Rightarrow \mathit{ReceivedSpec}$
PROOF BY $\mathit{SafeAcceptorIsAcceptor}$ DEF Init, $\mathit{ReceivedSpec}$

LEMMA $\mathit{ReceivedByLearnerSpecInit} \triangleq \mathit{Init} \Rightarrow \mathit{ReceivedByLearnerSpec}$
PROOF BY DEF Init, $\mathit{ReceivedByLearnerSpec}$, TypeOK

LEMMA $\mathit{ConnectedSpecInit} \triangleq \mathit{Init} \Rightarrow \mathit{ConnectedSpec}$
PROOF BY DEF Init, $\mathit{ConnectedSpec}$

```
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 $Spec$, $FullSafetyInvariant$
```