```
┌──────────────────────── MODULE HPaxos ────────────────────────┐
EXTENDS Integers, TLAPS, TLC

├────────────────────────────────────────────────────────────────┤
Ballot ≜ Nat

LEMMA BallotLeqTrans ≜
    ASSUME NEW A ∈ Ballot, NEW B ∈ Ballot, NEW C ∈ Ballot, A ≤ B, B ≤ CPROVE A ≤ C
PROOF BY DEF Ballot

LEMMA BallotLeLeqTrans ≜
    ASSUME NEW A ∈ Ballot, NEW B ∈ Ballot, NEW C ∈ Ballot, A < B, B ≤ CPROVE A < C
PROOF BY DEF Ballot

LEMMA BallotLeqLeTrans ≜
    ASSUME NEW A ∈ Ballot, NEW B ∈ Ballot, NEW C ∈ Ballot, A ≤ B, B < CPROVE A < C
PROOF BY DEF Ballot

LEMMA BallotLeNotLeq ≜ ASSUME NEW A ∈ Ballot, NEW B ∈ Ballot, A < BPROVE ¬B ≤ A
PROOF BY DEF Ballot

LEMMA BallotOrderCases ≜ ASSUME NEW A ∈ Ballot, NEW B ∈ BallotPROVE A < B ∨ B < A ∨ A = B
PROOF BY DEF Ballot

CONSTANT Value
ASSUME ValueNotEmpty ≜ Value ≠ {}

None ≜ CHOOSE v : v ∉ Value

├────────────────────────────────────────────────────────────────┤
CONSTANTS Acceptor,
          SafeAcceptor,
          FakeAcceptor,
          ByzQuorum,
          Learner

ASSUME SafeAcceptorAssumption ≜
          ∧ SafeAcceptor ∩ FakeAcceptor = {}
          ∧ SafeAcceptor ∪ FakeAcceptor = Acceptor

LEMMA SafeAcceptorIsAcceptor ≜ SafeAcceptor ⊆ Acceptor
PROOF BY SafeAcceptorAssumption

LEMMA FakeAcceptorIsAcceptor ≜ FakeAcceptor ⊆ Acceptor
PROOF BY SafeAcceptorAssumption

ASSUME BQAssumption ≜
          ∧ ∀ Q ∈ ByzQuorum : Q ⊆ Acceptor
```

```
ASSUME BallotAssumption ≜
          ∧ (Ballot ∪ { − 1}) ∩ Acceptor = {}
          ∧ (Ballot ∪ { − 1}) ∩ ByzQuorum = {}
          ∧ (Ballot ∪ { − 1}) ∩ Learner = {}

├──────────────────────────────────────────────────────────────────┤
 Learner graph

CONSTANT TrustLive
ASSUME TrustLiveAssumption ≜ TrustLive ∈ SUBSET [lr : Learner, q : ByzQuorum]

CONSTANT TrustSafe
ASSUME TrustSafeAssumption ≜ TrustSafe ∈ SUBSET [from : Learner, to : Learner, q : ByzQuorum]

ASSUME LearnerGraphAssumption ≜
          symmetry
          ∧ ∀ E ∈ TrustSafe :
              [from ↦ E.to, to ↦ E.from, q ↦ E.q] ∈ TrustSafe
          transitivity
          ∧ ∀ E1, E2 ∈ TrustSafe :
              E1.q = E2.q ∧ E1.to = E2.from ⇒
              [from ↦ E1.from, to ↦ E2.to, q ↦ E1.q] ∈ TrustSafe
          closure
          ∧ ∀ E ∈ TrustSafe : ∀ Q ∈ ByzQuorum :
              E.q ⊆ Q ⇒
              [from ↦ E.from, to ↦ E.to, q ↦ Q] ∈ TrustSafe
          validity
          ∧ ∀ E ∈ TrustSafe : ∀ Q1, Q2 ∈ ByzQuorum :
              [lr ↦ E.from, q ↦ Q1] ∈ TrustLive ∧
              [lr ↦ E.to, q ↦ Q2] ∈ TrustLive ⇒
              ∃ N ∈ E.q : N ∈ Q1 ∧ N ∈ Q2

 CONSTANT TrustWeak
 ASSUME TrustWeakAssumption ≜ TrustWeak ∈ SUBSET [lr : Learner, q : ByzQuorum]

 ASSUME WeakQuorumAssumption ≜

   ∧ ∀ L ∈ Learner : ∀ Q1, Q2 ∈ ByzQuorum :
       Q1 ⊆ Q2 ∧
       [lr ↦ L, q ↦ Q1] ∈ TrustWeak ⇒
       [lr ↦ L, q ↦ Q2] ∈ TrustWeak

   ∧ ∀ L ∈ Learner : ∀ WQ ∈ ByzQuorum :
       [lr ↦ L, q ↦ WQ] ∈ TrustWeak ⇒
       ∀ Q ∈ ByzQuorum :
           [from ↦ L, to ↦ L, q ↦ Q] ∈ TrustSafe ⇒
```

```
        ∃ N ∈ SafeAcceptor : N ∈ Q

CONSTANT  WeakQuorum
ASSUME  WeakQuorumIsByzQuorum ≜ WeakQuorum ⊆ ByzQuorum

ASSUME  WeakQuorumAssumption1 ≜
  ∀ WQ ∈ WeakQuorum : ∀ L ∈ Learner :
      [from ↦ L, to ↦ L, q ↦ WQ] ∈ TrustSafe ⇒
      ∃ S ∈ SafeAcceptor : S ∈ WQ

CONSTANT Ent
ASSUME EntanglementAssumption ≜
             ∧ Ent ∈ SUBSET (Learner × Learner)
             ∧ ∀ L1, L2 ∈ Learner :
                 ⟨L1, L2⟩ ∈ Ent ≡
                 [from ↦ L1, to ↦ L2, q ↦ SafeAcceptor] ∈ TrustSafe

LEMMA EntanglementSym ≜
    ASSUME NEW L1 ∈ Learner, NEW L2 ∈ Learner, ⟨L1, L2⟩ ∈ EntPROVE ⟨L2, L1⟩ ∈ Ent
PROOF BY EntanglementAssumption, LearnerGraphAssumption

LEMMA EntanglementSelf ≜
    ASSUME NEW L1 ∈ Learner, NEW L2 ∈ Learner, ⟨L1, L2⟩ ∈ EntPROVE ⟨L1, L1⟩ ∈ Ent
PROOF BY EntanglementAssumption, 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

LEMMA  EntanglementWeakQuorum ≜
  ASSUME  NEW  L1 ∈ Learner, NEW  L2 ∈ Learner,
       NEW  WQ ∈ WeakQuorum,
       ⟨L1, L2⟩ ∈ Ent
  PROVE ∃ N ∈ SafeAcceptor : N ∈ WQ
PROOF  BY  EntanglementAssumption, WeakQuorumAssumption

├─────────────────────────────────────────────────────────────┤
Messages

Message ≜
    [type : {"1a"}, lr : Learner, bal : Ballot] ∪
    [
        $type : \{\text{“1b”}\},$
        $lr \quad : Learner,$
        $acc \ \ : Acceptor,$
        $bal \ \ : Ballot,$
        $votes : \textsc{subset}\ [lr : Learner,\ bal : Ballot,\ val : Value],$
        $proposals : \textsc{subset}\ [lr : Learner,\ bal : Ballot,\ val : Value]$
    $] \cup$
    $[type : \{\text{“1c”}\},\ lr : Learner,\ bal : Ballot,\ val : Value] \cup$
    $[type : \{\text{“2av”}\},\ lr : Learner,\ acc : Acceptor,\ bal : Ballot,\ val : Value] \cup$
    $[type : \{\text{“2b”}\},\ lr : Learner,\ acc : Acceptor,\ bal : Ballot,\ val : Value]$
```

Algorithm specification

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

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

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

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

$KnowsSafeAt1(l,\ ac,\ b,\ v) \triangleq$
    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  $\exists c \in Ballot :$
          $\land\ c < b$
          $\land\ \exists BQ \in ByzQuorum :$
              $\land\ [lr \mapsto l, q \mapsto BQ] \in TrustLive$
              $\land\ \forall a \in BQ :$
                  $\exists m \in S :$
                      $\land\ m.acc = a$
                      $\land\ \forall p \in \{pp \in m.votes : \langle pp.lr, l \rangle \in connected[ac]\} :$
                          $\land\ p.bal \leq c$
                          $\land\ (p.bal = c) \Rightarrow (p.val = v)$
          $\land\ \exists WQ \in ByzQuorum :$
              $\land\ [lr \mapsto l, q \mapsto WQ] \in TrustLive$
              $\land\ \forall a \in WQ :$
                  $\exists m \in S :$
                      $\land\ m.acc = a$
                      $\land\ \exists p \in m.proposals :$
                          $\land\ p.lr = l$  NB differs from the ivy model
                          $\land\ p.bal = c$
                          $\land\ p.val = v$

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

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

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

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

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

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

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

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

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

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

$\mathit{Phase2b}(l, b, a, v) \triangleq$
    $\land\ \forall L \in \mathit{Learner} : \mathit{maxBal}[L, a] \leq b$
    $\land\ \exists Q \in \mathit{ByzQuorum} :$
```

```
            ∧ [lr ↦ l, q ↦ Q] ∈ TrustLive
            ∧ ∀ aa ∈ Q :
                ∃ m ∈ {mm ∈ received[a] :
                            ∧ mm.type = "2av"
                            ∧ mm.lr = l
                            ∧ mm.bal = b} :
                    ∧ m.val = v
                    ∧ m.acc = aa
    ∧   Send([type ↦ "2b", lr ↦ l, acc ↦ a, bal ↦ b, val ↦ v])
    ∧   votesSent′ = [votesSent EXCEPT ![a] =
                            votesSent[a] ∪ {[lr ↦ l, bal ↦ b, val ↦ v]}]
    ∧   UNCHANGED ⟨maxBal, 2avSent, received, connected, receivedByLearner, decision⟩

Recv(l, a) ≜
      ∧ ∃ m ∈ msgs : received′ = [received EXCEPT ![a] = received[a] ∪ {m}]
      ∧ UNCHANGED ⟨msgs, maxBal, 2avSent, votesSent, connected, receivedByLearner, decision⟩

Disconnect(a) ≜
    ∧ ∃ P ∈ SUBSET {LL ∈ Learner × Learner : LL ∉ Ent} :
        connected′ = [connected EXCEPT ![a] = connected[a] \ P]
    ∧ UNCHANGED ⟨msgs, maxBal, votesSent, 2avSent, received, receivedByLearner, decision⟩

FakeSend(a) ≜
    ∧ ∃ m ∈ {mm ∈ Message :
                    ∧ mm.acc = a
                    ∧ ∨ mm.type = "1b"
                      ∨ mm.type = "2av"
                      ∨ mm.type = "2b"} :
        Send(m)
   ∧ UNCHANGED ⟨maxBal, votesSent, 2avSent, received, connected, receivedByLearner, decision⟩

LearnerDecide(l, b) ≜
    ∧ ∃ v ∈ {vv ∈ Value : ChosenIn(l, b, vv)} :
        decision′ = [decision EXCEPT ![l, b] = decision[l, b] ∪ {v}]
    ∧ UNCHANGED ⟨msgs, maxBal, votesSent, 2avSent, received, connected, receivedByLearner⟩

LearnerRecv(l) ≜
    ∧ ∃ m ∈ {mm ∈ msgs : mm.type = "2b" ∧ mm.lr = l} :
        receivedByLearner′ =
            [receivedByLearner EXCEPT ![l] = receivedByLearner[l] ∪ {m}]
    ∧ UNCHANGED ⟨msgs, maxBal, votesSent, 2avSent, received, connected, decision⟩

ProposerAction ≜
    ∃ lrn ∈ Learner : ∃ proposer ∈ Ballot :
        ∨ Phase1a(lrn, proposer)
        ∨ ∃ v ∈ Value : Phase1c(lrn, proposer, v)
```

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

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

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

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

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

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

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

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

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

$LeftBallot(lr, acc, bal) \triangleq$
```

$$\begin{array}{l} \exists\, m \in \mathit{msgs} : \\ \quad \land\, m.\mathit{type} = \text{“1b”} \\ \quad \land\, m.\mathit{lr} = \mathit{lr} \\ \quad \land\, m.\mathit{acc} = \mathit{acc} \\ \quad \land\, \mathit{bal} < m.\mathit{bal} \end{array}$$

$$\mathit{ReceivedSpec} \;\triangleq\; \forall\, A \in \mathit{SafeAcceptor} : \mathit{received}[A] \subseteq \mathit{msgs}$$

$$\begin{array}{l} \mathit{ReceivedByLearnerSpec} \;\triangleq \\ \quad \land\, \mathit{receivedByLearner} \in [\mathit{Learner} \rightarrow \textsc{subset}\ \{\mathit{mm} \in \mathit{msgs} : \mathit{mm.type} = \text{“2b”}\}] \\ \quad \land\, \forall\, L \in \mathit{Learner} : \forall\, \mathit{mm} \in \mathit{Message} : \\ \qquad \mathit{mm} \in \mathit{receivedByLearner}[L] \Rightarrow \mathit{mm.lr} = L \end{array}$$

$$\begin{array}{l} \mathit{VotesSentSpec}1 \;\triangleq \\ \quad \forall\, A \in \mathit{SafeAcceptor} : \forall\, \mathit{vote} \in \mathit{votesSent}[A] : \mathit{VotedFor}(\mathit{vote.lr},\, A,\, \mathit{vote.bal},\, \mathit{vote.val}) \end{array}$$

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

$$\begin{array}{l} \mathit{VotesSentSpec}3 \;\triangleq \\ \quad \forall\, A \in \mathit{SafeAcceptor} : \forall\, B \in \mathit{Ballot} : \forall\, \mathit{vote} \in \mathit{votesSent}[A] : \\ \qquad \mathit{vote.bal} < B \Rightarrow \\ \qquad \exists\, P \in \mathit{votesSent}[A] : \\ \qquad\quad \mathit{MaxVote}(A,\, B,\, P) \land P.\mathit{lr} = \mathit{vote.lr} \land \mathit{vote.bal} \leq P.\mathit{bal} \end{array}$$

$$\begin{array}{l} \mathit{VotesSentSpec}4 \;\triangleq \\ \quad \forall\, A \in \mathit{SafeAcceptor} : \forall\, \mathit{vote}1,\, \mathit{vote}2 \in \mathit{votesSent}[A] : \\ \qquad \langle \mathit{vote}1.\mathit{lr},\, \mathit{vote}2.\mathit{lr} \rangle \in \mathit{Ent} \land \\ \qquad \mathit{vote}1.\mathit{bal} = \mathit{vote}2.\mathit{bal} \Rightarrow \mathit{vote}1.\mathit{val} = \mathit{vote}2.\mathit{val} \end{array}$$

$$2\mathit{avSentSpec}1 \;\triangleq\; \forall\, A \in \mathit{SafeAcceptor} : \forall\, p \in 2\mathit{avSent}[A] : \mathit{Proposed}(p.\mathit{lr},\, A,\, p.\mathit{bal},\, p.\mathit{val})$$

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

$$\begin{array}{l} 2\mathit{avSentSpec}3 \;\triangleq \\ \quad \forall\, L1,\, L2 \in \mathit{Learner} : \forall\, A \in \mathit{SafeAcceptor} : \forall\, B \in \mathit{Ballot} : \forall\, V1,\, V2 \in \mathit{Value} : \\ \qquad \langle L1,\, L2 \rangle \in \mathit{Ent} \land \\ \qquad [\mathit{lr} \mapsto L1,\, \mathit{bal} \mapsto B,\, \mathit{val} \mapsto V1] \in 2\mathit{avSent}[A] \land \\ \qquad [\mathit{lr} \mapsto L2,\, \mathit{bal} \mapsto B,\, \mathit{val} \mapsto V2] \in 2\mathit{avSent}[A] \Rightarrow V1 = V2 \end{array}$$

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

```
$DecisionSpec \triangleq$
    $\forall L \in Learner : \forall B \in Ballot : \forall V \in Value :$
        $V \in decision[L, B] \Rightarrow ChosenIn(L, B, V)$

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

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

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

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

```
LEMMA $MessageType \triangleq$
    ASSUME NEW $m \in Message$
    PROVE  $\land\ m.lr \in Learner$
           $\land\ m.bal \in Ballot$
           $\land\ (m.type = \text{"1b"} \lor m.type = \text{"2av"} \lor m.type = \text{"2b"}) \Rightarrow m.acc \in Acceptor$
```

```
            $\land (m.type = \text{"1c"} \lor m.type = \text{"2av"} \lor m.type = \text{"2b"}) \Rightarrow m.val \in Value$
            $\land (m.type = \text{"1b"}) \Rightarrow$
                  $\land\ m.votes \in$ SUBSET $[lr : Learner,\ bal : Ballot,\ val : Value]$
                  $\land\ m.proposals \in$ 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 $Phase1b(lrn,\ bal,\ acc)$
    $\langle 3 \rangle 1.$ $(votesSent \in [Acceptor \rightarrow$ SUBSET $[lr : Learner,\ bal : Ballot,\ val : Value]])'$
          BY $\langle 2 \rangle 1$  DEF $Phase1b,\ Phase2av,\ Phase2b,\ Send,\ TypeOK,\ Message$
    $\langle 3 \rangle 2.$ $(2avSent \in [Acceptor \rightarrow$ SUBSET $[lr : Learner,\ bal : Ballot,\ val : Value]])'$
         BY $\langle 2 \rangle 1$  DEF $Phase1b,\ Phase2av,\ Phase2b,\ Send,\ TypeOK,\ Message$
    $\langle 3 \rangle 3.$ $msgs' \in$ 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 $[lr : Learner,\ bal : Ballot,\ val : Value]])'$
```

```
            $\langle 4\rangle 0.$ $[lr \mapsto lrn, bal \mapsto bal, val \mapsto val] \in [lr : Learner, bal \quad : Ballot, val : Value]$
                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 $Message$
            $\langle 4\rangle 0.$ $[type \mapsto$ "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
```

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

LEMMA MaxBalMonotone ≜
    TypeOK ∧ Next ⇒ ∀ l ∈ Learner : ∀ a ∈ SafeAcceptor : maxBal[l, a] ≤ maxBal′[l, a]
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, NEW CONSTANT l ∈ Learner, NEW CONSTANT a ∈ SafeAcceptor
             PROVE  maxBal[l, a] ≤ maxBal′[l, a]
    OBVIOUS
⟨1⟩1.CASE ProposerAction
     BY ⟨1⟩1, SafeAcceptorIsAcceptor DEF ProposerAction, Phase1a, Phase1c, Send, TypeOK, Ballot
⟨1⟩2.CASE AcceptorSendAction
  ⟨2⟩ SUFFICES ASSUME NEW lrn ∈ 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  maxBal[l, a] ≤ (maxBal′)[l, a]
    BY ⟨1⟩2, SafeAcceptorIsAcceptor DEF AcceptorSendAction
  ⟨2⟩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 InitializedBallotInvariant ≜
  ∀ L ∈ Learner : ∀ B ∈ Ballot : Next ∧ InitializedBallot(L, B) ⇒ InitializedBallot(L, B)′
PROOF
⟨1⟩ SUFFICES ASSUME NEW L ∈ Learner, NEW B ∈ Ballot, Next, InitializedBallot(L, B)
          PROVE InitializedBallot(L, B)′
  OBVIOUS
⟨1⟩1.CASE ProposerAction BY ⟨1⟩1 DEF ProposerAction, Phase1a, Phase1c, Next
⟨1⟩2.CASE AcceptorSendAction BY ⟨1⟩2 DEF Phase1b, Phase2b, Phase2av, Next
⟨1⟩3.CASE AcceptorReceiveAction BY ⟨1⟩3 DEF AcceptorReceiveAction, Recv, Next
⟨1⟩4.CASE AcceptorDisconnectAction BY ⟨1⟩4 DEF AcceptorDisconnectAction, Disconnect, Next
⟨1⟩5.CASE LearnerAction BY ⟨1⟩5 DEF LearnerAction, LearnerRecv, LearnerDecide, Next
⟨1⟩6.CASE FakeAcceptorAction BY ⟨1⟩6 DEF FakeAcceptorAction, FakeSend, Send
⟨1⟩7. QED BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6 DEF Next

LEMMA 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 $![acc] =$
                                          $votesSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
                  PROVE $(\exists P \in votesSent[A] : MaxVote(A, B, P) \wedge P.lr = V.lr \wedge V.bal \leq P.bal)'$
        BY ⟨2⟩3 DEF $Phase2b$
    ⟨3⟩1.CASE $A = acc$
      ⟨4⟩0. DEFINE $v0 \triangleq [lr \mapsto lrn, bal \mapsto bal, val \mapsto val]$
      ⟨4⟩1. $v0 \in votesSent[A]'$BY ⟨3⟩1, ⟨1⟩0b DEF $TypeOK$
      ⟨4⟩2.CASE $V \in votesSent[A]$BY ⟨4⟩2
      ⟨4⟩3.CASE $V \notin votesSent[A]$
        ⟨5⟩0. $V = v0$BY ⟨4⟩3, ⟨3⟩1
        ⟨5⟩1.CASE $\forall P \in votesSent[A] : P.lr = lrn \Rightarrow P.bal \geq B$
          ⟨6⟩1. WITNESS $v0 \in votesSent[A]'$
          ⟨6⟩2. QED BY ⟨3⟩1, ⟨5⟩1, ⟨1⟩0b, ⟨5⟩0 DEF $Ballot$, $TypeOK$, $MaxVote$
        ⟨5⟩2.CASE $\exists P \in votesSent[A] : P.lr = lrn \wedge P.bal < B$
          ⟨6⟩1. PICK $P \in votesSent[A] : P.lr = lrn \wedge P.bal < B$BY ⟨5⟩2
          ⟨6⟩2. PICK $Pmax \in votesSent[A] : MaxVote(A, B, Pmax) \wedge Pmax.lr = lrn \wedge P.bal \leq Pmax.bal$BY ⟨6
          ⟨6⟩3. $Pmax \in votesSent[A]'$BY ⟨3⟩1, ⟨6⟩2
          ⟨6⟩4.CASE $Pmax.bal < bal$
            ⟨7⟩1. WITNESS $v0 \in votesSent[A]'$
            ⟨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 $bal \leq Pmax.bal$
            ⟨7⟩1. WITNESS $Pmax \in votesSent[A]'$
            ⟨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 $A \neq acc$BY ⟨3⟩2
    ⟨3⟩3. QED BY ⟨3⟩1, ⟨3⟩2
  ⟨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 $VotesSentSpec4Invariant \triangleq$
    $TypeOK \wedge Next \wedge MsgInv \wedge ReceivedSpec \wedge$
    $VotesSentSpec1 \wedge 2avSentSpec2 \wedge 2avSentSpec3 \wedge VotesSentSpec4 \Rightarrow$
    $VotesSentSpec4'$
PROOF
⟨1⟩ SUFFICES ASSUME $TypeOK$, $Next$, $MsgInv$, $ReceivedSpec$, $VotesSentSpec1$,
                    $2avSentSpec2$, $2avSentSpec3$, $VotesSentSpec4$,
                    NEW $A \in SafeAcceptor$,
                    NEW $vote1 \in votesSent'[A]$, NEW $vote2 \in votesSent'[A]$,
                    $\langle vote1.lr, vote2.lr \rangle \in Ent$,
                    $vote1.bal = vote2.bal$
            PROVE $vote1.val = vote2.val$
  BY DEF $VotesSentSpec4$
$\langle 1 \rangle$ USE DEF $MsgInv$
$\langle 1 \rangle$0a. $TypeOK$OBVIOUS
$\langle 1 \rangle$0b. $TypeOK'$BY $TypeOKInvariant$
$\langle 1 \rangle$1.CASE $ProposerAction$BY $\langle 1 \rangle 1$ DEF $ProposerAction$, $Phase1a$, $Phase1c$, $Send$, $VotesSentSpec4$
$\langle 1 \rangle$2.CASE $AcceptorSendAction$
  $\langle 2 \rangle$.SUFFICES ASSUME NEW $lrn \in Learner$,
                     NEW $bal \in Ballot$,
                     NEW $acc \in SafeAcceptor$,
                     NEW $val \in Value$,
                     $\lor Phase1b(lrn, bal, acc)$
                     $\lor Phase2av(lrn, bal, acc, val)$
                     $\lor Phase2b(lrn, bal, acc, val)$
              PROVE $vote1.val = vote2.val$
    BY $\langle 1 \rangle 2$ DEF $AcceptorSendAction$
  $\langle 2 \rangle$1.CASE $Phase1b(lrn, bal, acc)$BY $\langle 2 \rangle 1$ DEF $Phase1b$, $VotesSentSpec4$
  $\langle 2 \rangle$2.CASE $Phase2av(lrn, bal, acc, val)$BY $\langle 2 \rangle 2$ DEF $Phase2av$, $VotesSentSpec4$
  $\langle 2 \rangle$3.CASE $Phase2b(lrn, bal, acc, val)$
    $\langle 3 \rangle$ SUFFICES ASSUME $votesSent' = [votesSent$ EXCEPT $![acc] =$
                         $votesSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
               PROVE $vote1.val = vote2.val$
      BY $\langle 2 \rangle 3$ DEF $Phase2b$
    $\langle 3 \rangle$1.CASE $A = acc$
      $\langle 4 \rangle$1.CASE $vote1 \in votesSent[A] \land vote2 \in votesSent[A]$BY $\langle 4 \rangle 1$ DEF $VotesSentSpec4$
      $\langle 4 \rangle$2.CASE $vote1 \in votesSent[A] \land vote2 \notin votesSent[A]$
        $\langle 5 \rangle$0. $vote1.lr \in Learner \land vote1.val \in Value$BY $\langle 4 \rangle 2$ DEF $TypeOK$
        $\langle 5 \rangle$1. $vote2 = [lr \mapsto lrn, bal \mapsto bal, val \mapsto val]$BY $\langle 4 \rangle 2$
        $\langle 5 \rangle$2. PICK $Q2 \in ByzQuorum$ :
               $\land$ $[lr \mapsto lrn, q \mapsto Q2] \in TrustLive$
               $\land$ $\forall aa \in Q2$ :
                   $\exists m \in \{mm \in received[acc]$ :
                             $\land mm.type =$ "2av"
                             $\land mm.lr = lrn$
                             $\land mm.bal = bal\}$ :
                     $\land m.val = val$
                     $\land m.acc = aa$
             BY $\langle 5 \rangle 1$, $\langle 2 \rangle 3$ DEF $Phase2b$
        $\langle 5 \rangle$3. $\langle vote1.lr, lrn \rangle \in Ent \land vote1.bal = bal$BY $\langle 5 \rangle 1$
        $\langle 5 \rangle$4. PICK $m1 \in msgs$ :
               $\land$ $m1.type =$ "2b"
               $\land$ $m1.lr = vote1.lr$
        $\land\ m1.acc = A$
        $\land\ m1.bal = bal$
        $\land\ m1.val = vote1.val$
      BY $\langle 4\rangle 2$, $\langle 5\rangle 3$  DEF $VotesSentSpec1$, $VotedFor$
$\langle 5\rangle 5$. 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"
                        $\land\ m2av.lr = vote2.lr$
                        $\land\ m2av.acc = ba$
                        $\land\ m2av.bal\ = bal$
                        $\land\ m2av.val\ = vote2.val$
  $\langle 6\rangle 1.\ \exists\, Q2 \in \mathit{ByzQuorum} :$
             $\land\, [lr \mapsto m2.lr,\ q \mapsto Q2] \in \mathit{TrustLive}$
             $\land\, \forall\, ba \in Q2 :$
                 $\exists\, m2av \in \mathit{received}[m2.acc] :$
                     $\land\, m2av.type = \text{“2av”}$
                     $\land\, m2av.lr = m2.lr$
                     $\land\, m2av.acc = ba$
                     $\land\, m2av.bal\ = m2.bal$
                     $\land\, m2av.val\ = m2.val$
       BY $\langle 5\rangle 4,\ \langle 3\rangle 1$  DEF $\mathit{MsgInv2b},\ \mathit{TypeOK}$
  $\langle 6\rangle 2.$ QED BY $\langle 5\rangle 4,\ \langle 6\rangle 1,\ \langle 3\rangle 1$
$\langle 5\rangle 6.\ \langle lrn,\ vote2.lr\rangle \in \mathit{Ent}$BY $\langle 5\rangle 3$
$\langle 5\rangle 7.$ PICK $S \in \mathit{SafeAcceptor} : S \in Q1 \land S \in Q2$BY $\langle 5\rangle 2,\ \langle 5\rangle 5,\ \langle 5\rangle 6,\ \langle 5\rangle 0,\ \mathit{EntanglementTrustLive}$
$\langle 5\rangle 8.$ PICK $m2av2 \in \mathit{received}[acc] :$
               $\land\, m2av2.type = \text{“2av”}$
               $\land\, m2av2.lr = vote2.lr$
               $\land\, m2av2.acc = S$
               $\land\, m2av2.bal\ = bal$
               $\land\, m2av2.val\ = vote2.val$
      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 $\langle 5\rangle 8,\ \langle 5\rangle 0,\ \mathit{SafeAcceptorIsAcceptor}$ DEF $\mathit{ReceivedSpec},\ \mathit{TypeOK}$
$\langle 5\rangle 10.\ [lr \mapsto vote2.lr,\ bal \mapsto bal,\ val \mapsto vote2.val] \in 2avSent[S]$
        BY $\langle 5\rangle 9,\ \langle 5\rangle 0$  DEF $2avSentSpec2,\ \mathit{Proposed}$
$\langle 5\rangle 11.$ PICK $m2av1 \in \mathit{received}[acc] :$
               $\land\, m2av1.type = \text{“2av”}$
               $\land\, m2av1.lr = lrn$
               $\land\, m2av1.acc = S$
               $\land\, m2av1.bal\ = bal$
               $\land\, m2av1.val\ = val$
        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 $\langle 5\rangle 11,\ \mathit{SafeAcceptorIsAcceptor}$ DEF $\mathit{ReceivedSpec},\ \mathit{TypeOK}$
$\langle 5\rangle 13.\ [lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val] \in 2avSent[S]$
        BY $\langle 5\rangle 12,\ \mathit{SafeAcceptorIsAcceptor}$ DEF $2avSentSpec2,\ \mathit{Proposed}$
        $\langle 5\rangle 14.$ $vote2.val = val$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 $Phase2b$, $Send$
```

```
          ⟨5⟩3. WITNESS ⟨5⟩2
          ⟨5⟩10. QED BY ⟨3⟩2, ⟨4⟩3
        ⟨4⟩10. QED BY ⟨4⟩2, ⟨4⟩3
    ⟨3⟩3.CASE acc ≠ ABY ⟨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 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 2avSentSpec2Invariant ≜ Next ∧ 2avSentSpec2 ⇒ 2avSentSpec2′
PROOF
⟨1⟩ SUFFICES ASSUME Next, 2avSentSpec2,
                    NEW L ∈ Learner, NEW A ∈ SafeAcceptor, NEW B ∈ Ballot, NEW V ∈ Value,
                    Proposed(L, A, B, V)′
             PROVE ([lr ↦ L, bal ↦ B, val ↦ V] ∈ 2avSent[A])′
    BY  DEF 2avSentSpec2
⟨1⟩ USE  DEF 2avSentSpec2
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Next, 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 ([lr ↦ L, bal ↦ B, val ↦ V] ∈ 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 ↦ "2av", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val]),
                        2avSent′ = [2avSent EXCEPT ![acc] =
                                     2avSent[acc] ∪ {[lr ↦ lrn, bal ↦ bal, val ↦ val]}]
                 PROVE ([lr ↦ L, bal ↦ B, val ↦ V] ∈ 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 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
```

```
$\langle 1\rangle 6.$CASE $\mathit{FakeAcceptorAction}$BY $\langle 1\rangle 6$  DEF $\mathit{FakeAcceptorAction}, \mathit{FakeSend}, \mathit{Send}$
$\langle 1\rangle 7.$ 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 $\mathit{Next}$

LEMMA $2avSentSpec3Invariant \triangleq \mathit{Next} \land \mathit{ConnectedSpec} \land 2avSentSpec3 \Rightarrow 2avSentSpec3'$
PROOF
$\langle 1\rangle$ SUFFICES ASSUME $\mathit{Next}, \mathit{ConnectedSpec}, 2avSentSpec3,$
                 NEW $L1 \in \mathit{Learner}$, NEW $L2 \in \mathit{Learner}$, NEW $A \in \mathit{SafeAcceptor}$, NEW $B \in \mathit{Ballot}$,
                 NEW $V1 \in \mathit{Value}$, NEW $V2 \in \mathit{Value}$,
                 $\langle L1, L2\rangle \in \mathit{Ent}$,
                 $[lr \mapsto L1, bal \mapsto B, val \mapsto V1] \in 2avSent'[A]$,
                 $[lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent'[A]$
                 PROVE $V1 = V2$
    BY  DEF $2avSentSpec3$
$\langle 1\rangle$ USE  DEF $2avSentSpec3$
$\langle 1\rangle 1.$CASE $\mathit{ProposerAction}$BY $\langle 1\rangle 1$  DEF $\mathit{ProposerAction}, \mathit{Phase1a}, \mathit{Phase1c}, \mathit{Next}, \mathit{Send}$
$\langle 1\rangle 2.$CASE $\mathit{AcceptorSendAction}$
  $\langle 2\rangle$ SUFFICES ASSUME NEW $lrn \in \mathit{Learner}$,
                             NEW $bal \in \mathit{Ballot}$,
                             NEW $acc \in \mathit{SafeAcceptor}$,
                             NEW $val \in \mathit{Value}$,
                             $\lor\ \mathit{Phase1b}(lrn, bal, acc)$
                             $\lor\ \mathit{Phase2av}(lrn, bal, acc, val)$
                             $\lor\ \mathit{Phase2b}(lrn, bal, acc, val)$
                        PROVE $V1 = V2$
      BY $\langle 1\rangle 2$  DEF $\mathit{AcceptorSendAction}$
  $\langle 2\rangle 1.$CASE $\mathit{Phase1b}(lrn, bal, acc)$BY $\langle 2\rangle 1$  DEF $\mathit{Phase1b}$
  $\langle 2\rangle 2.$CASE $\mathit{Phase2av}(lrn, bal, acc, val)$
    $\langle 3\rangle$ SUFFICES
           ASSUME NEW $v \in \mathit{Value}$,
                  $\forall P \in \{p \in 2avSent[acc] : p.bal = bal \land \langle p.lr, lrn\rangle \in connected[acc]\} : P.val = v$,
                  $2avSent' = [2avSent$ EXCEPT $![acc] =$
                             $2avSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto v]\}]$
           PROVE $V1 = V2$
        BY $\langle 2\rangle 2$  DEF $\mathit{Phase2av}$
    $\langle 3\rangle 1.$CASE $A = acc$
      $\langle 4\rangle 1.$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 $\langle 4\rangle 1, \langle 3\rangle 1$
      $\langle 4\rangle 3.$CASE $\land\ [lr \mapsto L1, bal \mapsto B, val \mapsto V1] \notin 2avSent[A]$
                $\land\ [lr \mapsto L2, bal \mapsto B, val \mapsto V2] \in 2avSent[A]$
        $\langle 5\rangle 1.$ $\langle L2, L1\rangle \in \mathit{Ent}$BY $\mathit{EntanglementSym}$
        $\langle 5\rangle 2.$ QED BY $\langle 4\rangle 3, \langle 3\rangle 1, \langle 5\rangle 1$  DEF $\mathit{ConnectedSpec}$
      $\langle 4\rangle 2.$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 $\langle 4\rangle 2, \langle 3\rangle 1$  DEF $\mathit{ConnectedSpec}$
      ⟨4⟩4.CASE  ∧ [lr ↦ L1, bal ↦ B, val ↦ V1] ∉ 2avSent[A]
                 ∧ [lr ↦ L2, bal ↦ B, val ↦ V2] ∉ 2avSent[A]
           BY ⟨4⟩4, ⟨3⟩1
      ⟨4⟩5. QED BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3, ⟨4⟩4
    ⟨3⟩2.CASE A ≠ accBY ⟨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 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 DecisionSpecInvariant ≜ TypeOK ∧ Next ∧ DecisionSpec ⇒ DecisionSpec′
PROOF
⟨1⟩ SUFFICES ASSUME Next, TypeOK, DecisionSpec,
                    NEW L ∈ Learner, NEW B ∈ Ballot, NEW V ∈ Value,
                    V ∈ decision′[L, B]
                    PROVE  ChosenIn(L, B, V)′
    BY  DEF DecisionSpec
⟨1⟩ USE  DEF DecisionSpec
⟨1⟩1.CASE ProposerActionBY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Next, Send
⟨1⟩2.CASE AcceptorSendActionBY ⟨1⟩2  DEF AcceptorSendAction, Phase1b, Phase2av, Phase2b, Next, Send
⟨1⟩3.CASE AcceptorReceiveActionBY ⟨1⟩3  DEF AcceptorReceiveAction, Recv, Next
⟨1⟩4.CASE AcceptorDisconnectActionBY ⟨1⟩4  DEF AcceptorDisconnectAction, Disconnect, Next
⟨1⟩5.CASE LearnerAction
  ⟨2⟩ SUFFICES ASSUME NEW lrn ∈ Learner, NEW bal ∈ Ballot,
                      ∨ LearnerDecide(lrn, bal)
                      ∨ LearnerRecv(lrn)
               PROVE  ChosenIn(L, B, V)′
       BY ⟨1⟩5  DEF LearnerAction
  ⟨2⟩2.CASE LearnerDecide(lrn, bal)
    ⟨3⟩0a. TypeOKOBVIOUS
    ⟨3⟩0b. TypeOK′BY TypeOKInvariant
    ⟨3⟩1.CASE V ∈ decision[L, B]BY ⟨3⟩1, ⟨2⟩2  DEF ChosenIn, LearnerDecide
    ⟨3⟩2.CASE V ∉ 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 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 ConnectedSpecInvariant ≜ Next ∧ ConnectedSpec ⇒ ConnectedSpec′
```

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

LEMMA $MsgInvInvariant \triangleq$
    $TypeOK \land MsgInv \land VotesSentSpec1 \land VotesSentSpec2 \land VotesSentSpec3 \land 2avSentSpec1 \land$
    $Next \Rightarrow MsgInv'$
PROOF
$\langle 1 \rangle$ USE  DEF $MsgInv$
$\langle 1 \rangle 1$b. ASSUME $TypeOK$, $VotesSentSpec1$, $VotesSentSpec2$, $VotesSentSpec3$, $2avSentSpec1$, $Next$,
        $\forall\, m \in msgs : m.acc \in SafeAcceptor \land m.type = \text{“1b”} \Rightarrow MsgInv1b(m)$,
        NEW $m \in msgs'$, $m.acc \in SafeAcceptor$, $m.type = \text{“1b”}$
        PROVE $MsgInv1b(m)'$
  $\langle 2 \rangle 0$. $TypeOK$BY $\langle 1 \rangle 1$b
  $\langle 2 \rangle 0$a. $TypeOK'$BY $\langle 1 \rangle 1$b, $TypeOKInvariant$
  $\langle 2 \rangle 0$b. $m \in Message$BY $\langle 2 \rangle 0$a  DEF $TypeOK$
  $\langle 2 \rangle 0$c. $maxBal \in [Learner \times Acceptor \rightarrow Ballot]$BY $\langle 1 \rangle 1$b  DEF $TypeOK$
  $\langle 2 \rangle 0$d. $maxBal' \in [Learner \times Acceptor \rightarrow Ballot]$BY $\langle 2 \rangle 0$a  DEF $TypeOK$
  $\langle 2 \rangle 0$e. $m.type = \text{“1b”}$BY $\langle 1 \rangle 1$b
  $\langle 2 \rangle 0$f. $m.bal \in Ballot$BY $\langle 2 \rangle 0$b, $\langle 2 \rangle 0$e  DEF $Message$, $Ballot$
  $\langle 2 \rangle 0$g. $maxBal[m.lr, m.acc] \in Ballot$BY $\langle 2 \rangle 0$b, $\langle 2 \rangle 0$c, $\langle 2 \rangle 0$e  DEF $Message$
  $\langle 2 \rangle 0$h. $maxBal'[m.lr, m.acc] \in Ballot$BY $\langle 2 \rangle 0$b, $\langle 2 \rangle 0$d, $\langle 2 \rangle 0$e  DEF $Message$
  $\langle 2 \rangle 0$i. $maxBal[m.lr, m.acc] \leq maxBal'[m.lr, m.acc]$BY $\langle 1 \rangle 1$b, $\langle 2 \rangle 0$b, $MaxBalMonotone$ DEF $TypeOK$, $Messa$
  $\langle 2 \rangle 1$.CASE $ProposerAction$
    $\langle 3 \rangle$ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $proposer \in Ballot$, NEW $val \in Value$,
                              $\lor\, Phase1a(lrn, proposer)$
                              $\lor\, Phase1c(lrn, proposer, val)$
                     PROVE $MsgInv1b(m)'$
      BY $\langle 2 \rangle 1$, $ValueNotEmpty$ DEF $ProposerAction$
    $\langle 3 \rangle 1$.CASE $Phase1a(lrn, proposer)$
      $\langle 4 \rangle 1$. $m \in msgs$BY $\langle 3 \rangle 1$, $\langle 2 \rangle 0$e  DEF $Phase1a$, $Send$
      $\langle 4 \rangle 2$. QED BY $\langle 1 \rangle 1$b, $\langle 4 \rangle 1$, $\langle 3 \rangle 1$  DEF $Phase1a$, $MsgInv1b$
    $\langle 3 \rangle 2$.CASE $Phase1c(lrn, proposer, val)$
    ⟨4⟩1. $m \in msgs$BY ⟨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 $Phase1b$, $Send$
  $\langle 5\rangle 3.$ QED BY $\langle 5\rangle 1$, $\langle 5\rangle 2$
 $\langle 4\rangle 10.$ QED BY $\langle 4\rangle 1$, $\langle 4\rangle 5$, $\langle 4\rangle 6$ DEF $MsgInv1b$
$\langle 3\rangle 2.$CASE $Phase2av(lrn, bal, acc, val)$
 $\langle 4\rangle$ SUFFICES
     ASSUME $maxBal[lrn, acc] \leq bal$,
            $Send([type \mapsto \text{“2av”}, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val])$,
            $2avSent' = [2avSent$ 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 $TypeOK$
      $\langle 4\rangle$2. $(m.bal \ \leq maxBal[m.lr,\ m.acc])'$BY $\langle 4\rangle$1, $\langle 1\rangle$1b, $\langle 3\rangle$3  DEF $Phase2b$, $MsgInv1b$
      $\langle 4\rangle$4. $(m.proposals = \{p \in 2avSent[m.acc] : p.bal < m.bal \land p.lr = m.lr\})'$
            BY $\langle 4\rangle$1, $\langle 1\rangle$1b, $\langle 3\rangle$3  DEF $Phase2b$, $MsgInv1b$
      $\langle 4\rangle$5. $(m.votes = \{p \ \in votesSent[m.acc] : MaxVote(m.acc,\ m.bal,\ p)\})'$
        $\langle 5\rangle$1.CASE $m.acc \neq acc$
          $\langle 6\rangle$1. $votesSent'[m.acc] = votesSent[m.acc]$BY $\langle 3\rangle$3, $\langle 4\rangle$1, $\langle 5\rangle$1, $\langle 2\rangle$0, $\langle 2\rangle$0e, $MessageType$ DEF $Phas$
          $\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$2. $maxBal[m.lr,\ m.acc] \leq bal$BY $\langle 2\rangle$0a, $\langle 2\rangle$0e, $\langle 5\rangle$2, $MessageType$ DEF $Ballot$, $TypeOK$
          $\langle 6\rangle$3. $m.bal \in Ballot$BY $\langle 2\rangle$0a, $MessageType$ DEF $TypeOK$
          $\langle 6\rangle$4. $m.bal \leq bal$BY $\langle 6\rangle$1, $\langle 6\rangle$2, $\langle 6\rangle$3, $\langle 2\rangle$0g, $BallotLeqTrans$
          $\langle 6\rangle$5. SUFFICES $\{p \in votesSent[acc] : MaxVote(acc,\ m.bal,\ p)\} =$
                        $\{p \in votesSent'[acc] : MaxVote(acc,\ m.bal,\ p)'\}$
              BY $\langle 4\rangle$1, $\langle 2\rangle$0e, $\langle 1\rangle$1b, $\langle 5\rangle$2  DEF $MsgInv1b$
          $\langle 6\rangle$6. $\{p \in votesSent[acc] : MaxVote(acc,\ m.bal,\ p)\} \subseteq$
                $\{p \in votesSent'[acc] : MaxVote(acc,\ m.bal,\ p)'\}$
            BY $\langle 4\rangle$1a, $\langle 2\rangle$0, $VotesSentMonotone$, $\langle 6\rangle$4  DEF $TypeOK$
          $\langle 6\rangle$7. $\{p \in votesSent'[acc] : MaxVote(acc,\ m.bal,\ p)'\} \subseteq$
                $\{p \in votesSent[acc] : MaxVote(acc,\ m.bal,\ p)\}$
            $\langle 7\rangle$1. SUFFICES ASSUME NEW $p \in votesSent'[acc]$,
                                        $MaxVote(acc,\ m.bal,\ p)'$,
                                        $p \notin votesSent[acc]$
                                PROVE FALSE
                  OBVIOUS
            $\langle 7\rangle$2. $p = [lr \mapsto lrn,\ bal \mapsto bal,\ val \mapsto val]$BY $\langle 7\rangle$1, $\langle 5\rangle$2, $\langle 2\rangle$0  DEF $TypeOK$
            $\langle 7\rangle$3. QED BY $\langle 7\rangle$2, $\langle 7\rangle$1, $\langle 6\rangle$4, $\langle 6\rangle$3, $BallotLeNotLeq$ DEF $MaxVote$
          $\langle 6\rangle$8. QED BY $\langle 6\rangle$6, $\langle 6\rangle$7
        $\langle 5\rangle$3. QED BY $\langle 5\rangle$1, $\langle 5\rangle$2
      $\langle 4\rangle$6. QED BY $\langle 4\rangle$2, $\langle 4\rangle$4, $\langle 4\rangle$5  DEF $MsgInv1b$
    $\langle 3\rangle$4. QED BY $\langle 3\rangle$1, $\langle 3\rangle$2, $\langle 3\rangle$3
  $\langle 2\rangle$4.CASE $AcceptorReceiveAction$BY $\langle 1\rangle$1b, $\langle 2\rangle$4  DEF $AcceptorReceiveAction$, $Recv$, $MsgInv1b$, $Next$
  $\langle 2\rangle$5.CASE $AcceptorDisconnectAction$BY $\langle 1\rangle$1b, $\langle 2\rangle$5  DEF $AcceptorDisconnectAction$, $Disconnect$, $MsgInv1b$,
  $\langle 2\rangle$6.CASE $LearnerAction$BY $\langle 1\rangle$1b, $\langle 2\rangle$6  DEF $LearnerAction$, $LearnerRecv$, $LearnerDecide$, $MsgInv1b$, $Next$
  $\langle 2\rangle$7.CASE $FakeAcceptorAction$BY $\langle 1\rangle$1b, $\langle 2\rangle$7, $SafeAcceptorAssumption$ DEF $FakeAcceptorAction$, $FakeSend$,
  $\langle 2\rangle$8. QED BY $\langle 1\rangle$1b, $\langle 2\rangle$0a, $\langle 2\rangle$1, $\langle 2\rangle$2, $\langle 2\rangle$4, $\langle 2\rangle$5, $\langle 2\rangle$6, $\langle 2\rangle$7  DEF $Next$
$\langle 1\rangle$2av. ASSUME $TypeOK$, $Next$,
                $\forall\, m \in msgs : m.acc \in SafeAcceptor \land m.type = \text{“2av”} \Rightarrow MsgInv2av(m)$,
                NEW $m \in msgs'$, $m.acc \in SafeAcceptor$, $m.type = \text{“2av”}$
        PROVE $MsgInv2av(m)'$
  $\langle 2\rangle$0a. $TypeOK$BY $\langle 1\rangle$2av
  $\langle 2\rangle$0b. $TypeOK'$BY $\langle 1\rangle$2av, $TypeOKInvariant$
  $\langle 2\rangle$0e. $m.type = \text{“2av”}$BY $\langle 1\rangle$2av
  $\langle 2\rangle$1.CASE $ProposerAction$
  ⟨3⟩0. m ∈ msgsBY ⟨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, A
      ⟨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
```

```
        $\langle 5\rangle 4.$ $AnnouncedValue(m.lr, m.bal, m.val)'$BY $\langle 5\rangle 1$
        $\langle 5\rangle 5.$ $KnowsSafeAt(m.lr, m.acc, m.bal, m.val)'$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 $AcceptorReceiveAction$, $Recv$
    $\langle 3\rangle 6.$ $(\exists\, Q \in ByzQuorum :$
                ∧ [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)′
        ⟨7⟩1. PICK Q0 ∈ ByzQuorum :
                  ∧  [lr ↦ m.lr, q ↦ Q0] ∈ TrustLive
                  ∧  ∀ ba ∈ Q0 :
                        ∃ m1b ∈ received[m.acc] :
                          ∧ m1b.type = "1b"
                          ∧ m1b.lr = m.lr
                          ∧ m1b.acc = ba
                          ∧ m1b.bal = m.bal
              BY ⟨1⟩2av, ⟨3⟩1, ⟨2⟩0e  DEF MsgInv2av
        ⟨7⟩2. WITNESS Q0 ∈ 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 AcceptorDisconnectActionBY ⟨1⟩2av, ⟨2⟩5  DEF AcceptorDisconnectAction, Disconnect, MsgInv2a
  ⟨2⟩6.CASE LearnerActionBY ⟨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, ∀ m ∈ msgs : m.acc ∈ SafeAcceptor ∧ m.type = "2b" ⇒ MsgInv2b(m),
         NEW m ∈ msgs′, m.acc ∈ SafeAcceptor, m.type = "2b"
         PROVE MsgInv2b(m)′
  ⟨2⟩0a. TypeOKBY ⟨1⟩2b
  ⟨2⟩0b. TypeOK′BY ⟨1⟩2b, TypeOKInvariant
  ⟨2⟩0c. m ∈ MessageBY ⟨2⟩0b  DEF TypeOK
  ⟨2⟩0d. m.acc ∈ SafeAcceptorBY ⟨1⟩2b
  ⟨2⟩0e. m.type = "2b"BY ⟨1⟩2b
  ⟨2⟩1.CASE ProposerAction
    ⟨3⟩1. m ∈ msgsBY ⟨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 ∈ 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 $MsgInv2b(m)'$
    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 = $ "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 = $ "2av"
                                  $\land mm.lr = lrn$
                                  $\land mm.bal = bal\} :$
                       $\land m\_1.val = val$
                       $\land m\_1.acc = aa$,
               $Send([type \mapsto $ "2b"$, lr \mapsto lrn, acc \mapsto acc, bal \mapsto bal, val \mapsto val])$,
               $votesSent' = [votesSent$ EXCEPT $![acc] =$
                                  $votesSent[acc] \cup \{[lr \mapsto lrn, bal \mapsto bal, val \mapsto val]\}]$
               PROVE $MsgInv2b(m)'$
        BY $\langle 3\rangle 3$ DEF $Phase2b$
    $\langle 5\rangle 1.$ $m = [type \mapsto $ "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 $TypeOK$
    $\langle 5\rangle 3.$ $(\exists Q\_1 \in ByzQuorum :$
          $\land [lr \mapsto m.lr, q \mapsto Q\_1] \in TrustLive$
```

```
                $\land \forall ba \in Q\_1 :$
                    $\exists m2av \in received[m.acc] :$
                      $\land m2av.type = \text{"2av"}$
                      $\land m2av.lr = m.lr$
                      $\land m2av.acc = ba$
                      $\land m2av.bal = m.bal$
                      $\land m2av.val = m.val)'$
          $\langle 6 \rangle 1.$ WITNESS $Q \in ByzQuorum$
          $\langle 6 \rangle 2.$ QED BY $\langle 5 \rangle 1$, $\langle 5 \rangle 1e$, $\langle 2 \rangle 0a$  DEF $Send$, $TypeOK$
        $\langle 5 \rangle 4.$ QED BY $\langle 5 \rangle 2$, $\langle 5 \rangle 3$  DEF $MsgInv2b$
      $\langle 4 \rangle 3.$ QED BY $\langle 4 \rangle 1$, $\langle 4 \rangle 2$
    $\langle 3 \rangle 5.$ QED BY $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, $\langle 3 \rangle 3$
  $\langle 2 \rangle 4.$CASE $AcceptorReceiveAction$
    $\langle 3 \rangle 0.$ SUFFICES ASSUME NEW $lrn \in Learner$,
                                NEW $acc \in Acceptor$,
                                NEW $m0 \in msgs$,
                                $received' = [received$ EXCEPT $![acc] = received[acc] \cup \{m0\}]$,
                                UNCHANGED $\langle msgs, maxBal, 2avSent, votesSent, connected,$
                                            $receivedByLearner, decision \rangle$
                         PROVE  $MsgInv2b(m)'$
      BY $\langle 2 \rangle 4$, $\langle 2 \rangle 0b$  DEF $AcceptorReceiveAction$, $Recv$, $TypeOK$
    $\langle 3 \rangle 2.$ $m \in msgs$BY $\langle 3 \rangle 0$, $\langle 1 \rangle 2b$
    $\langle 3 \rangle 2a.$ $m \in Message$BY $\langle 3 \rangle 2$, $\langle 1 \rangle 2b$  DEF $TypeOK$
    $\langle 3 \rangle 2b.$ $TypeOK$BY $\langle 1 \rangle 2b$  DEF $Phase2b$
    $\langle 3 \rangle 2c.$ $TypeOK'$BY $\langle 1 \rangle 2b$, $\langle 3 \rangle 2b$, $TypeOKInvariant$
    $\langle 3 \rangle 3.$ $[lr \mapsto m.lr, bal \mapsto m.bal, val \mapsto m.val] \in votesSent'[m.acc]$BY $\langle 3 \rangle 0$, $\langle 1 \rangle 2b$  DEF $MsgInv2b$
    $\langle 3 \rangle 5.$ PICK $Q0 \in ByzQuorum :$
            $\land [lr \mapsto m.lr, q \mapsto Q0] \in TrustLive$
            $\land \forall ba \in Q0 :$
                $\exists m2av \in received[m.acc] :$
                  $\land m2av.type = \text{"2av"}$
                  $\land m2av.lr = m.lr$
                  $\land m2av.acc = ba$
                  $\land m2av.bal = m.bal$
                  $\land m2av.val = m.val$BY $\langle 1 \rangle 2b$, $\langle 2 \rangle 0e$, $\langle 3 \rangle 2$  DEF $MsgInv2b$
    $\langle 3 \rangle 7.$ $(\exists Q \in ByzQuorum :$
            $\land [lr \mapsto m.lr, q \mapsto Q] \in TrustLive$
            $\land \forall ba \in Q :$
                $\exists m2av \in received[m.acc] :$
                  $\land m2av.type = \text{"2av"}$
                  $\land m2av.lr = m.lr$
                  $\land m2av.acc = ba$
                  $\land m2av.bal = m.bal$
                  $\land m2av.val = m.val)'$
      $\langle 4 \rangle 0.$ WITNESS $Q0 \in ByzQuorum$
```

```
        ⟨4⟩1. QED BY ⟨1⟩2b, ⟨3⟩5, ⟨3⟩2b, ⟨3⟩2c, MessageType, ReceivedMonotone DEF TypeOK
    ⟨3⟩8. QED BY ⟨3⟩3, ⟨3⟩7  DEF MsgInv2b
  ⟨2⟩5.CASE AcceptorDisconnectActionBY ⟨1⟩2b, ⟨2⟩5  DEF AcceptorDisconnectAction, Disconnect, MsgInv2b,
  ⟨2⟩6.CASE LearnerActionBY ⟨1⟩2b, ⟨2⟩6  DEF LearnerAction, LearnerRecv, LearnerDecide, MsgInv2b, Next
  ⟨2⟩7.CASE FakeAcceptorAction
        BY ⟨1⟩2b, ⟨2⟩7, SafeAcceptorAssumption DEF FakeAcceptorAction, FakeSend, MsgInv2b, Send
  ⟨2⟩8. QED BY ⟨1⟩2b, ⟨2⟩0a, ⟨2⟩1, ⟨2⟩2, ⟨2⟩4, ⟨2⟩5, ⟨2⟩6, ⟨2⟩7  DEF Next
⟨1⟩3. QED BY ⟨1⟩1b, ⟨1⟩2av, ⟨1⟩2b

CannotDecide(Q, L, B, V) ≜
    ∃ A ∈ SafeAcceptor :
        ∧ A ∈ Q
        ∧ ∃ L0 ∈ Learner : LeftBallot(L0, A, B)  TODO: check if used
        ∧ ¬VotedFor(L, A, B, V)

HeterogeneousSpec ≜
    ∀ L1, L2   ∈ Learner :
    ∀ B1, B2  ∈ Ballot :
    ∀ V1, V2 ∈ Value :
    ∀ A2 ∈ SafeAcceptor :
    ∀ Q  ∈ ByzQuorum :
    ∀ M ∈ msgs :
        ∧ ⟨L1, L2⟩ ∈ Ent
        ∧ [lr ↦ L1, q ↦ Q] ∈ TrustLive
        ∧ M.type = "2av" ∧ M.lr = L2 ∧ M.acc = A2 ∧ M.bal = B2 ∧ M.val = V2
        ∧ B1 < B2
        ∧ V1 ≠ V2
        ⇒
        CannotDecide(Q, L1, B1, V1)

LEMMA HeterogeneousSpecInvariant ≜
    TypeOK ∧ Next ∧ ReceivedSpec ∧
    2avSentSpec1 ∧
    VotesSentSpec2 ∧ VotesSentSpec3 ∧ VotesSentSpec4 ∧
    ConnectedSpec ∧ MsgInv ∧
    HeterogeneousSpec ⇒ HeterogeneousSpec′
PROOF
⟨1⟩ SUFFICES ASSUME TypeOK, Next, ReceivedSpec, 2avSentSpec1, VotesSentSpec2, VotesSentSpec3, VotesSe
                    ConnectedSpec, MsgInv, HeterogeneousSpec,
                    NEW L1  ∈ Learner, NEW L2 ∈ Learner,
                    NEW B1  ∈ Ballot, NEW B2 ∈ Ballot,
                    NEW V1 ∈ Value, NEW V2 ∈ Value,
                    NEW A2  ∈ SafeAcceptor,
                    NEW Q1 ∈ ByzQuorum,
                    NEW m ∈ msgs′,
                    ⟨L1, L2⟩ ∈ Ent,
                        [lr ↦ L1, q ↦ Q1] ∈ TrustLive,
                        m.type = "2av", m.lr = L2, m.acc = A2, m.bal = B2, m.val = V2,
                        B1 < B2,
                        V1 ≠ V2
                PROVE  CannotDecide(Q1, L1, B1, V1)′
    BY  DEF HeterogeneousSpec
⟨1⟩ USE  DEF MsgInv
⟨1⟩0a. TypeOK OBVIOUS
⟨1⟩0b. TypeOK′ BY TypeOKInvariant
⟨1⟩0c. m ∈ Message BY ⟨1⟩0b  DEF TypeOK
⟨1⟩1.CASE ProposerAction BY ⟨1⟩1  DEF ProposerAction, Phase1a, Phase1c, Next, Send, HeterogeneousSpec
⟨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  CannotDecide(Q1, L1, B1, V1)′
      BY ⟨1⟩2  DEF AcceptorSendAction
  ⟨2⟩1.CASE Phase1b(lrn, bal, acc)
    ⟨3⟩1. m ∈ msgs BY ⟨2⟩1, ⟨1⟩0b  DEF Phase1b, Send, TypeOK, Message
    ⟨3⟩2. QED BY ⟨3⟩1  DEF HeterogeneousSpec
  ⟨2⟩2.CASE Phase2av(lrn, bal, acc, val)
    ⟨3⟩0. msgs ⊆ msgs′ BY ⟨2⟩2  DEF Phase2av, Send
    ⟨3⟩1.CASE m ∈ msgs BY ⟨3⟩1  DEF HeterogeneousSpec
    ⟨3⟩2.CASE m ∉ msgs
      ⟨4⟩0. m = [type ↦ "2av", lr ↦ lrn, acc ↦ acc, bal ↦ bal, val ↦ val]
            BY ⟨3⟩2, ⟨2⟩2  DEF Phase2av, Send
      ⟨4⟩0a. lrn = L2 ∧ acc = A2 ∧ bal = B2 ∧ val = V2 BY ⟨4⟩0
      ⟨4⟩1. maxBal[L2, A2] ≤ B2 BY ⟨2⟩2, ⟨4⟩0a  DEF Phase2av
      ⟨4⟩2. KnowsSafeAt(L2, A2, B2, V2) BY ⟨2⟩2, ⟨4⟩0a  DEF Phase2av
      ⟨4⟩3a.CASE KnowsSafeAt1(L2, A2, B2, V2)
        ⟨5⟩0. USE  DEF CannotDecide
        ⟨5⟩1. PICK Q2 ∈ ByzQuorum :
                ∧ [lr ↦ L2, q ↦ Q2] ∈ TrustLive
                ∧ ∀ a ∈ Q2 :
                    ∃ m1b ∈ received[A2] :
                       ∧  m1b.type = "1b"
                       ∧  m1b.lr = L2
                       ∧  m1b.bal = B2
                       ∧  m1b.acc = a
                       ∧  ∀ p ∈ {pp ∈ m1b.votes : ⟨pp.lr, L2⟩ ∈ connected[A2]} :
                             B2 ≤ 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 $\langle 5\rangle 2$, $\langle 5\rangle 6$, $\langle 5\rangle 7$
$\langle 4\rangle 3b.$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 ⟨4⟩3b, ⟨4⟩0a  DEF KnowsSafeAt2, Ballot
⟨5⟩2. PICK $S1 \in SafeAcceptor : S1 \in Q1 \land S1 \in BQ$BY EntanglementTrustLive, ⟨4⟩0, ⟨5⟩1
⟨5⟩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 ⟨5⟩1, ⟨5⟩2
⟨5⟩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 ⟨5⟩4, SafeAcceptorIsAcceptor DEF TypeOK, ReceivedSpec
⟨5⟩6.CASE $\neg VotedFor(L1,\ S1,\ B1,\ V1)$
  ⟨6⟩1. $\neg VotedFor(L1,\ S1,\ B1,\ V1)'$BY ⟨5⟩6, ⟨2⟩2  DEF VotedFor, Phase2av, Send
  ⟨6⟩2. QED BY ⟨6⟩1, ⟨5⟩2, ⟨5⟩5, MsgsMonotone DEF LeftBallot, CannotDecide
⟨5⟩7.CASE $VotedFor(L1,\ S1,\ B1,\ V1)$
  ⟨6⟩1. $[lr \mapsto L1,\ bal \mapsto B1,\ val \mapsto V1] \in votesSent[S1]$BY ⟨5⟩7  DEF VotesSentSpec2
  ⟨6⟩2. PICK $P \in votesSent[S1] : MaxVote(S1,\ B2,\ P) \land P.lr = L1 \land B1 \leq P.bal$
    ⟨7⟩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 ⟨6⟩1
    ⟨7⟩2. QED BY ⟨7⟩1, SafeAcceptorIsAcceptor DEF VotesSentSpec3, TypeOK
  ⟨6⟩3. $P \in m1.votes$BY ⟨6⟩2, ⟨5⟩5  DEF MsgInv1b
  ⟨6⟩4. $\langle P.lr,\ L2\rangle \in connected[A2]$BY ⟨5⟩5, ⟨6⟩2  DEF ConnectedSpec
  ⟨6⟩5. $P.bal \in Ballot$BY ⟨5⟩5, ⟨6⟩3, SafeAcceptorIsAcceptor, MessageType DEF TypeOK
  ⟨6⟩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 $\langle 4\rangle 3$a, $\langle 4\rangle 3$b, $\langle 4\rangle 2$  DEF $KnowsSafeAt$
```

```
    ⟨3⟩3. QED BY ⟨3⟩1, ⟨3⟩2
  ⟨2⟩3.CASE $Phase2b(lrn, bal, acc, val)$
    ⟨3⟩1. $m \in msgs$BY ⟨2⟩3, ⟨1⟩0b  DEF $Phase2b$, $Send$, $TypeOK$
    ⟨3⟩2. QED BY ⟨3⟩1  DEF $HeterogeneousSpec$
  ⟨2⟩4. QED BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3.CASE $AcceptorReceiveAction$BY ⟨1⟩3  DEF $AcceptorReceiveAction$, $Next$, $Recv$, $HeterogeneousSpec$
⟨1⟩4.CASE $AcceptorDisconnectAction$BY ⟨1⟩4  DEF $AcceptorDisconnectAction$, $Disconnect$, $Next$, $Heterogeneou$
⟨1⟩5.CASE $LearnerAction$
  ⟨2⟩ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$,
                        $\lor LearnerDecide(lrn, bal)$
                        $\lor LearnerRecv(lrn)$
                 PROVE  $CannotDecide(Q1, L1, B1, V1)'$
     BY ⟨1⟩5  DEF $LearnerAction$
  ⟨2⟩2.CASE $LearnerDecide(lrn, bal)$BY ⟨2⟩2  DEF $LearnerDecide$, $Next$, $HeterogeneousSpec$
  ⟨2⟩3.CASE $LearnerRecv(lrn)$BY ⟨2⟩2  DEF $LearnerRecv$, $Next$, $HeterogeneousSpec$
  ⟨2⟩4. QED BY ⟨2⟩2, ⟨2⟩3
⟨1⟩6.CASE $FakeAcceptorAction$BY ⟨1⟩6, $SafeAcceptorAssumption$ DEF $FakeAcceptorAction$, $FakeSend$, $Send$, $F$
⟨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 ∈ receivedByLearner[L1] : m1.acc = A ∧ m1.bal = B ∧ m1.val = V1BY ⟨1⟩1, ⟨1⟩3  DEF Chosen
⟨1⟩5. PICK m2 ∈ receivedByLearner[L2] : m2.acc = A ∧ m2.bal = B ∧ m2.val = V2BY ⟨1⟩2, ⟨1⟩3  DEF Chosen
⟨1⟩6. ∧ m1 ∈ msgs
      ∧ m1.type = "2b"
      ∧ m1.lr = L1
      ∧ m1.acc = A
      ∧ m1.bal  = B
      ∧ m1.val  = V1
      BY ⟨1⟩4  DEF ReceivedByLearnerSpec, TypeOK
⟨1⟩7. ∧ m2 ∈ msgs
      ∧ m2.type = "2b"
      ∧ m2.lr = L2
      ∧ m2.acc = A
      ∧ m2.bal  = B
      ∧ m2.val  = V2
      BY ⟨1⟩5  DEF ReceivedByLearnerSpec, TypeOK
⟨1⟩8. [lr ↦ L1, bal ↦ B, val ↦ V1] ∈ votesSent[A]BY ⟨1⟩6  DEF MsgInv2b
⟨1⟩9. [lr ↦ L2, bal ↦ B, val ↦ V2] ∈ votesSent[A]BY ⟨1⟩7  DEF MsgInv2b
⟨1⟩100. QED BY ⟨1⟩8, ⟨1⟩9  DEF VotesSentSpec4

LEMMA ChosenSafeCaseLt ≜
   ASSUME NEW L1 ∈ Learner, NEW L2 ∈ Learner,
                NEW B1 ∈ Ballot, NEW B2 ∈ Ballot,
                NEW V1 ∈ Value, NEW V2 ∈ Value,
                TypeOK, ReceivedSpec, ReceivedByLearnerSpec, MsgInv,
                HeterogeneousSpec,
                ⟨L1, L2⟩ ∈ Ent,
                B1 < B2,
                ChosenIn(L1, B1, V1), ChosenIn(L2, B2, V2)
   PROVE  V1 = V2
PROOF
⟨1⟩ USE  DEF MsgInv
⟨1⟩ SUFFICES ASSUME  V1 ≠ V2PROVE  FALSEOBVIOUS
⟨1⟩1. PICK  Q1 ∈ ByzQuorum :
         ∧  [lr ↦ L1, q ↦ Q1] ∈ TrustLive
         ∧  ∀ aa ∈ Q1 :
              ∃ m ∈ {mm ∈ receivedByLearner[L1] : mm.bal = B1} :
                 ∧ m.val  = V1
                 ∧ m.acc  = aa
      BY  DEF ChosenIn
⟨1⟩2. PICK  Q2 ∈ ByzQuorum :
         ∧  [lr ↦ L2, q ↦ Q2] ∈ TrustLive
         ∧  ∀ aa ∈ Q2 :
              ∃ m ∈ {mm ∈ receivedByLearner[L2] : mm.bal = B2} :
                 ∧ m.val  = V2
                $\land\ m.acc\ = aa$
    BY  DEF $ChosenIn$
⟨1⟩3. PICK $A \in SafeAcceptor : A \in Q1 \land A \in Q2$BY $EntanglementTrustLive$, ⟨1⟩1, ⟨1⟩2
⟨1⟩4. PICK $m1 \in receivedByLearner[L1] : m1.acc = A \land m1.bal = B1 \land m1.val = V1$BY ⟨1⟩1, ⟨1⟩3  DEF $Chose$
⟨1⟩5. PICK $m2 \in receivedByLearner[L2] : m2.acc = A \land m2.bal = B2 \land m2.val = V2$BY ⟨1⟩2, ⟨1⟩3  DEF $Chose$
⟨1⟩6. $\land\ m1 \in msgs$
      $\land\ m1.type =$ "2b"
      $\land\ m1.lr = L1$
      $\land\ m1.acc = A$
      $\land\ m1.bal\ = B1$
      $\land\ m1.val\ = V1$
      BY ⟨1⟩4  DEF $ReceivedByLearnerSpec$, $TypeOK$
⟨1⟩7. $\land\ m2 \in msgs$
      $\land\ m2.type =$ "2b"
      $\land\ m2.lr = L2$
      $\land\ m2.acc = A$
      $\land\ m2.bal\ = B2$
      $\land\ m2.val\ = V2$
      BY ⟨1⟩5  DEF $ReceivedByLearnerSpec$, $TypeOK$
⟨1⟩10. PICK $R1 \in ByzQuorum$ :
            $\land\ [lr \mapsto L1,\ q \mapsto R1] \in TrustLive$
         $\land\ \forall\ aa \in R1$ :
           $\exists\ m2av \in received[L1,\ A]$ :
             $\land\ m2av.type =$ "2av"
             $\land\ m2av.acc = aa$
             $\land\ m2av.bal = B1$
             $\land\ m2av.val\ = V1$
       BY ⟨1⟩6  DEF $MsgInv2b$
⟨1⟩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 ⟨1⟩7  DEF $MsgInv2b$
⟨1⟩12. PICK $A0 \in SafeAcceptor : A0 \in R1 \land A0 \in R2$BY $EntanglementTrustLive$, ⟨1⟩10, ⟨1⟩11
⟨1⟩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 ⟨1⟩12, ⟨1⟩11
⟨1⟩16. $\land\ m2av2 \in msgs$
       $\land\ m2av2.type =$ "2av"
       $\land\ m2av2.lr = L2$
```

```
        ∧ m2av2.acc = A0
        ∧ m2av2.bal = B2
        ∧ m2av2.val = V2
        BY ⟨1⟩14, SafeAcceptorIsAcceptor DEF ReceivedSpec, TypeOK
⟨1⟩17. CannotDecide(Q1, L1, B1, V1)
  ⟨2⟩1. [lr ↦ L1, q ↦ Q1] ∈ TrustLiveBY ⟨1⟩1
  ⟨2⟩5. QED BY ⟨1⟩16, ⟨2⟩1  DEF HeterogeneousSpec
⟨1⟩18. PICK S ∈ SafeAcceptor : S ∈ Q1 ∧ ¬VotedFor(L1, S, B1, V1)BY ⟨1⟩17  DEF CannotDecide
⟨1⟩19. PICK m ∈ receivedByLearner[L1] : m.acc = S ∧ m.bal = B1 ∧ m.val = V1
         BY ⟨1⟩18, ⟨1⟩1  DEF CannotDecide
⟨1⟩20. ∧ m ∈ {mm ∈ msgs : mm.type = "2b"}
        ∧ m.lr = L1
        ∧ m.acc = S
        ∧ m.bal = B1
        ∧ m.val = V1
        BY ⟨1⟩19  DEF ReceivedByLearnerSpec, TypeOK
⟨1⟩50. QED BY ⟨1⟩20, ⟨1⟩18  DEF CannotDecide, VotedFor, ReceivedByLearnerSpec, TypeOK

LEMMA ChosenSafe ≜
    ASSUME NEW L1 ∈ Learner, NEW L2 ∈ Learner,
                   NEW B1 ∈ Ballot, NEW B2 ∈ Ballot,
                   NEW V1 ∈ Value, NEW V2 ∈ Value,
                   TypeOK, ReceivedSpec, ReceivedByLearnerSpec, VotesSentSpec4, MsgInv,
                   HeterogeneousSpec,
                   ⟨L1, L2⟩ ∈ Ent,
                   ChosenIn(L1, B1, V1), ChosenIn(L2, B2, V2)
    PROVE V1 = V2
PROOF
⟨1⟩ USE  DEF MsgInv
⟨1⟩1. PICK Q1 ∈ ByzQuorum :
          ∧  [lr ↦ L1, q ↦ Q1] ∈ TrustLive
          ∧  ∀ aa ∈ Q1 :
                ∃ m ∈ {mm ∈ receivedByLearner[L1] : mm.bal = B1} :
                    ∧ m.val  = V1
                    ∧ m.acc  = aa
       BY  DEF ChosenIn
⟨1⟩2. PICK Q2 ∈ ByzQuorum :
          ∧  [lr ↦ L2, q ↦ Q2] ∈ TrustLive
          ∧  ∀ aa ∈ Q2 :
                ∃ m ∈ {mm ∈ receivedByLearner[L2] : mm.bal = B2} :
                    ∧ m.val  = V2
                    ∧ m.acc  = aa
       BY  DEF ChosenIn
⟨1⟩3. PICK A ∈ SafeAcceptor : A ∈ Q1 ∧ A ∈ Q2BY EntanglementTrustLive, ⟨1⟩1, ⟨1⟩2
⟨1⟩4. PICK m1 ∈ receivedByLearner[L1] : m1.acc = A ∧ m1.bal = B1 ∧ m1.val = V1BY ⟨1⟩1, ⟨1⟩3  DEF 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} = $ "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 $\mathit{ReceivedByLearnerSpec}, \mathit{TypeOK}$
$\langle 1 \rangle 7.$ $\land\ m2 \in \mathit{msgs}$
      $\land\ m2.\mathit{type} = $ "2b"
      $\land\ m2.\mathit{lr} = L2$
      $\land\ m2.\mathit{acc} = A$
      $\land\ m2.\mathit{bal} = B2$
      $\land\ m2.\mathit{val} = V2$
    BY $\langle 1 \rangle 5$ DEF $\mathit{ReceivedByLearnerSpec}, \mathit{TypeOK}$
$\langle 1 \rangle 8.$ $[\mathit{lr} \mapsto L1,\ \mathit{bal} \mapsto B1,\ \mathit{val} \mapsto V1] \in \mathit{votesSent}[A]$BY $\langle 1 \rangle 6$ DEF $\mathit{MsgInv2b}$
$\langle 1 \rangle 9.$ $[\mathit{lr} \mapsto L2,\ \mathit{bal} \mapsto B2,\ \mathit{val} \mapsto V2] \in \mathit{votesSent}[A]$BY $\langle 1 \rangle 7$ DEF $\mathit{MsgInv2b}$
$\langle 1 \rangle 10.$ PICK $R1 \in \mathit{ByzQuorum} :$
            $\land\ [\mathit{lr} \mapsto L1,\ q \mapsto R1] \in \mathit{TrustLive}$
            $\land\ \forall\, aa \in R1 :$
                $\exists\, m2av \in \mathit{received}[A] :$
                    $\land\ m2av.\mathit{type} = $ "2av"
                    $\land\ m2av.\mathit{lr} = L1$
                    $\land\ m2av.\mathit{acc} = aa$
                    $\land\ m2av.\mathit{bal} = B1$
                    $\land\ m2av.\mathit{val} = V1$
      BY $\langle 1 \rangle 6$ DEF $\mathit{MsgInv2b}$
$\langle 1 \rangle 11.$ PICK $R2 \in \mathit{ByzQuorum} :$
            $\land\ [\mathit{lr} \mapsto L2,\ q \mapsto R2] \in \mathit{TrustLive}$
            $\land\ \forall\, aa \in R2 :$
                $\exists\, m2av \in \mathit{received}[A] :$
                    $\land\ m2av.\mathit{type} = $ "2av"
                    $\land\ m2av.\mathit{lr} = L2$
                    $\land\ m2av.\mathit{acc} = aa$
                    $\land\ m2av.\mathit{bal} = B2$
                    $\land\ m2av.\mathit{val} = V2$
      BY $\langle 1 \rangle 7$ DEF $\mathit{MsgInv2b}$
$\langle 1 \rangle 12.$ PICK $A0 \in \mathit{SafeAcceptor} : A0 \in R1 \land A0 \in R2$BY $\mathit{EntanglementTrustLive}, \langle 1 \rangle 10, \langle 1 \rangle 11$
$\langle 1 \rangle 13.$ PICK $m2av1 \in \mathit{received}[A] :$
            $m2av1.\mathit{type} = $ "2av" $\land\ m2av1.\mathit{lr} = L1 \land m2av1.\mathit{acc} = A0 \land m2av1.\mathit{bal} = B1 \land m2av1.\mathit{val} = V1$
      BY $\langle 1 \rangle 12, \langle 1 \rangle 10$
$\langle 1 \rangle 14.$ PICK $m2av2 \in \mathit{received}[A] :$
            $m2av2.\mathit{type} = $ "2av" $\land\ m2av2.\mathit{lr} = L2 \land m2av2.\mathit{acc} = A0 \land m2av2.\mathit{bal} = B2 \land m2av2.\mathit{val} = V2$
      BY $\langle 1 \rangle 12, \langle 1 \rangle 11$
$\langle 1 \rangle 15.$ $\land\ m2av1 \in \mathit{msgs}$
```

```
        ∧ m2av1.type = "2av"
        ∧ m2av1.lr = L1
        ∧ m2av1.acc = A0
        ∧ m2av1.bal = B1
        ∧ m2av1.val = V1
       BY ⟨1⟩13, SafeAcceptorIsAcceptor DEF ReceivedSpec, TypeOK
⟨1⟩16. ∧ m2av2 ∈ 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 ≜  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 $bal \in Ballot$,
                              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 $\langle 1\rangle 2$ DEF $AcceptorSendAction$
    $\langle 2\rangle 2$.CASE $Phase1b(lrn, bal, acc)$BY $\langle 2\rangle 2$, $\langle 1\rangle 0a$, $\langle 1\rangle 0b$ DEF $AcceptorSendAction$, $Send$, $Phase1b$, $Safety$, $Ty$
    $\langle 2\rangle 3$.CASE $Phase2av(lrn, bal, acc, val)$BY $\langle 2\rangle 3$, $\langle 1\rangle 0a$, $\langle 1\rangle 0b$ DEF $AcceptorSendAction$, $Send$, $Phase2av$, $Saf$
    $\langle 2\rangle 4$.CASE $Phase2b(lrn, bal, acc, val)$BY $\langle 2\rangle 4$, $\langle 1\rangle 0a$, $\langle 1\rangle 0b$ DEF $AcceptorSendAction$, $Send$, $Phase2b$, $Safety$
    $\langle 2\rangle 5$. QED BY $\langle 2\rangle 2$, $\langle 2\rangle 3$, $\langle 2\rangle 4$
  $\langle 1\rangle 3$.CASE $AcceptorReceiveAction$BY $\langle 1\rangle 3$, $\langle 1\rangle 0a$, $\langle 1\rangle 0b$ DEF $AcceptorReceiveAction$, $Recv$, $TypeOK$, $Safety$
  $\langle 1\rangle 4$.CASE $AcceptorDisconnectAction$BY $\langle 1\rangle 4$ DEF $AcceptorDisconnectAction$, $Disconnect$, $Safety$
  $\langle 1\rangle 5$.CASE $LearnerAction$
    $\langle 2\rangle$ SUFFICES ASSUME NEW $lrn \in Learner$, NEW $bal \in Ballot$,
                              $\lor$ $LearnerDecide(lrn, bal)$
                              $\lor$ $LearnerRecv(lrn)$
                  PROVE $V1 = V2$BY $\langle 1\rangle 5$ DEF $LearnerAction$
    $\langle 2\rangle 1$.CASE $LearnerRecv(lrn)$BY $\langle 2\rangle 1$ DEF $LearnerRecv$, $Safety$
    $\langle 2\rangle 2$.CASE $LearnerDecide(lrn, bal)$
      $\langle 3\rangle$ SUFFICES ASSUME NEW $val \in Value$,
                              $ChosenIn(lrn, bal, val)$,
                              $decision' = [decision$ EXCEPT $![\langle lrn, bal\rangle] = decision[lrn, bal] \cup \{val\}]$,
                              UNCHANGED $\langle msgs, maxBal, votesSent, 2avSent, received, connected, receivedByLe$
                    PROVE $V1 = V2$
        BY $\langle 2\rangle 2$ DEF $LearnerDecide$
      $\langle 3\rangle 0$.CASE $V1 = V2$BY $\langle 3\rangle 0$
      $\langle 3\rangle 1$.CASE $V1 \neq V2$
        $\langle 4\rangle 1$.CASE $val \neq V1 \land val \neq V2$BY $\langle 4\rangle 1$ DEF $Safety$, $TypeOK$
        $\langle 4\rangle 2$.CASE $val = V1$
          $\langle 5\rangle 0$. $V2 \in decision[L2, B2]$BY $\langle 3\rangle 1$, $\langle 4\rangle 2$ DEF $TypeOK$
          $\langle 5\rangle 1$. $ChosenIn(L2, B2, V2)$BY $\langle 5\rangle 0$ DEF $DecisionSpec$
          $\langle 5\rangle 2$.CASE $V1 \in decision[L1, B1]$BY $\langle 5\rangle 0$, $\langle 5\rangle 2$ DEF $Safety$
          $\langle 5\rangle 3$.CASE $V1 \notin decision[L1, B1]$
            $\langle 6\rangle 1$. $lrn = L1 \land bal = B1$BY $\langle 5\rangle 3$, $\langle 4\rangle 2$ DEF $TypeOK$
            $\langle 6\rangle 2$. $ChosenIn(L1, B1, V1)$BY $\langle 6\rangle 1$, $\langle 4\rangle 2$
            $\langle 6\rangle 3$. QED BY $\langle 5\rangle 1$, $\langle 6\rangle 2$, $ChosenSafe$
          $\langle 5\rangle 4$. QED BY $\langle 5\rangle 2$, $\langle 5\rangle 3$
        $\langle 4\rangle 3$.CASE $val = V2$
          $\langle 5\rangle 0$. $V1 \in decision[L1, B1]$BY $\langle 3\rangle 1$, $\langle 4\rangle 3$ DEF $TypeOK$
          $\langle 5\rangle 1$. $ChosenIn(L1, B1, V1)$BY $\langle 5\rangle 0$ DEF $DecisionSpec$
          $\langle 5\rangle 2$.CASE $V2 \in decision[L2, B2]$BY $\langle 5\rangle 0$, $\langle 5\rangle 2$ DEF $Safety$
          $\langle 5\rangle 3$.CASE $V2 \notin decision[L2, B2]$
            $\langle 6\rangle 1$. $lrn = L2 \land bal = B2$BY $\langle 5\rangle 3$, $\langle 4\rangle 3$ DEF $TypeOK$
            $\langle 6\rangle 2.$ $ChosenIn(L2, B2, V2)$BY $\langle 6\rangle 1$, $\langle 4\rangle 3$
            $\langle 6\rangle 10.$ QED BY $\langle 5\rangle 1$, $\langle 6\rangle 2$, $ChosenSafe$
          $\langle 5\rangle 4.$ QED BY $\langle 5\rangle 2$, $\langle 5\rangle 3$
        $\langle 4\rangle 4.$ QED BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$
      $\langle 3\rangle 2.$ QED BY $\langle 3\rangle 0$, $\langle 3\rangle 1$
  $\langle 2\rangle 3.$ QED BY $\langle 2\rangle 1$, $\langle 2\rangle 2$
$\langle 1\rangle 6.$CASE $FakeAcceptorAction$BY $\langle 1\rangle 6$  DEF $FakeAcceptorAction$, $FakeSend$, $Send$, $Safety$
$\langle 1\rangle 7.$ QED BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$, $\langle 1\rangle 4$, $\langle 1\rangle 5$, $\langle 1\rangle 6$  DEF $Next$
```

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

```
LEMMA $TypeOKInit \triangleq Init \Rightarrow TypeOK$
PROOF BY  DEF $Init$, $TypeOK$

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

LEMMA $2avSentSpec1Init \triangleq Init \Rightarrow 2avSentSpec1$
PROOF BY  DEF $Init$, $2avSentSpec1$

LEMMA $2avSentSpec2Init \triangleq Init \Rightarrow 2avSentSpec2$
PROOF BY  DEF $Init$, $2avSentSpec2$, $Proposed$

LEMMA $2avSentSpec3Init \triangleq Init \Rightarrow 2avSentSpec3$
PROOF BY  DEF $Init$, $2avSentSpec3$, $TypeOK$

LEMMA $VotesSentSpec1Init \triangleq Init \Rightarrow VotesSentSpec1$
PROOF BY  DEF $Init$, $VotesSentSpec1$

LEMMA $VotesSentSpec2Init \triangleq Init \Rightarrow VotesSentSpec2$
PROOF BY  DEF $Init$, $VotesSentSpec2$, $VotedFor$

LEMMA $VotesSentSpec3Init \triangleq Init \Rightarrow VotesSentSpec3$
PROOF BY  DEF $Init$, $VotesSentSpec3$

LEMMA $VotesSentSpec4Init \triangleq Init \Rightarrow VotesSentSpec4$
PROOF BY  DEF $Init$, $VotesSentSpec4$
```

```
LEMMA ReceivedSpecInit ≜ Init ⇒ ReceivedSpec
PROOF BY SafeAcceptorIsAcceptor DEF Init, ReceivedSpec

LEMMA ReceivedByLearnerSpecInit ≜ Init ⇒ ReceivedByLearnerSpec
PROOF BY  DEF Init, ReceivedByLearnerSpec, TypeOK

LEMMA ConnectedSpecInit ≜ Init ⇒ ConnectedSpec
PROOF BY  DEF Init, ConnectedSpec

LEMMA DecisionSpecInit ≜ Init ⇒ DecisionSpec
PROOF BY  DEF Init, DecisionSpec

LEMMA HeterogeneousSpecInit ≜ Init ⇒ HeterogeneousSpec
PROOF BY  DEF Init, HeterogeneousSpec

LEMMA SafetyInit ≜ Init ⇒ Safety
PROOF BY  DEF Init, Safety

LEMMA FullSafetyInvariantInit ≜ Init ⇒ FullSafetyInvariant
PROOF BY TypeOKInit, MsgInvInit,
            2avSentSpec1Init, 2avSentSpec2Init, 2avSentSpec3Init,
            VotesSentSpec1Init, VotesSentSpec2Init, VotesSentSpec3Init, VotesSentSpec4Init,
            ReceivedSpecInit, ReceivedByLearnerSpecInit, ConnectedSpecInit, DecisionSpecInit,
            HeterogeneousSpecInit, SafetyInit
         DEF FullSafetyInvariant

LEMMA FullSafetyInvariantNext ≜ FullSafetyInvariant ∧ [Next]_vars ⇒ FullSafetyInvariant′
PROOF
⟨1⟩ SUFFICES ASSUME FullSafetyInvariant, [Next]_varsPROVE FullSafetyInvariant′OBVIOUS
⟨1⟩1.CASE NextBY ⟨1⟩1,
       TypeOKInvariant, MsgInvInvariant,
       2avSentSpec1Invariant, 2avSentSpec2Invariant, 2avSentSpec3Invariant,
       VotesSentSpec1Invariant, VotesSentSpec2Invariant, VotesSentSpec3Invariant, VotesSentSpec4Invarian
       ReceivedSpecInvariant, ReceivedByLearnerSpecInvariant, ConnectedSpecInvariant, DecisionSpecInvaria
       HeterogeneousSpecInvariant, SafetyStep
      DEF FullSafetyInvariant
⟨1⟩2.CASE vars = vars′BY ⟨1⟩2  DEF vars, FullSafetyInvariant, TypeOK, MsgInv,
        2avSentSpec1, 2avSentSpec2, 2avSentSpec3,
        VotesSentSpec1, VotesSentSpec2, VotesSentSpec3, VotesSentSpec4,
        ReceivedSpec, ReceivedByLearnerSpec, ConnectedSpec, DecisionSpec,
        MsgInv1b, MsgInv2av, MsgInv2b,
        Safety
⟨1⟩3. QED BY ⟨1⟩1, ⟨1⟩2

THEOREM SafetyResult ≜ Spec ⇒ □Safety
PROOF BY PTL, FullSafetyInvariantInit, FullSafetyInvariantNext DEF Spec, FullSafetyInvariant
```