```
1  ---------- MODULE Paxos ----------
   Specification and Verification of Basic Paxos.
   See http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html ≠ paxos-simple
7  EXTENDS Integers, TLAPS, TLC
8  ----------
9  CONSTANTS Acceptors, Values, Quorums

11 ASSUME QuorumAssumption ≜
12            ∧ Quorums ⊆ SUBSET Acceptors
13            ∧ ∀ Q1, Q2 ∈ Quorums : Q1 ∩ Q2 ≠ {}

15 LEMMA QuorumNonEmpty ≜ ∀ Q ∈ Quorums : Q ≠ {}
16 BY QuorumAssumption

18 Ballots ≜ Nat

20 None ≜ CHOOSE v : v ∉ Values

22 LEMMA NoneNotAValue ≜ None ∉ Values
23 BY NoSetContainsEverything DEF None

25 Messages ≜      [type : {"1a"}, bal : Ballots]
26            ∪    [type : {"1b"}, bal : Ballots, maxVBal : Ballots ∪ {−1},
27                  maxVal : Values ∪ {None}, acc : Acceptors]
28            ∪    [type : {"2a"}, bal : Ballots, val : Values]
29            ∪    [type : {"2b"}, bal : Ballots, val : Values, acc : Acceptors]
30 ----------
31 VARIABLES msgs,     the set of messages that have been sent.
32           maxBal,   maxBal[a]: the highest-number ballot acceptor a has participated in.
33           maxVBal,  maxVBal[a]: the highest ballot in which a has voted;
34           maxVal    maxVal[a]: the value it voted for in that ballot.

36 vars ≜ ⟨msgs, maxBal, maxVBal, maxVal⟩

38 TypeOK ≜ ∧ msgs ∈ SUBSET Messages
39          ∧ maxVBal ∈ [Acceptors → Ballots ∪ {−1}]
40          ∧ maxBal  ∈ [Acceptors → Ballots ∪ {−1}]
41          ∧ maxVal  ∈ [Acceptors → Values ∪ {None}]
42          ∧ ∀ a ∈ Acceptors : maxBal[a] ≥ maxVBal[a]

44 Send(m) ≜ msgs′ = msgs ∪ {m}
45 ----------
46 Init ≜ ∧ msgs = {}
47        ∧ maxVBal = [a ∈ Acceptors ↦ −1]
48        ∧ maxBal  = [a ∈ Acceptors ↦ −1]
49        ∧ maxVal  = [a ∈ Acceptors ↦ None]

51 Phase1a(b) ≜ ∧ ¬∃ m ∈ msgs : (m.type = "1a") ∧ (m.bal = b)
```

```
52                  ∧ Send([type ↦ "1a", bal ↦ b])
53                  ∧ UNCHANGED ⟨maxVBal, maxBal, maxVal⟩

55  Phase1b(a) ≜
56    ∃ m ∈ msgs :
57      ∧ m.type = "1a"
58      ∧ m.bal > maxBal[a]
59      ∧ maxBal' = [maxBal EXCEPT ![a] = m.bal]
60      ∧ Send([type  ↦ "1b", bal ↦ m.bal,
61              maxVBal ↦ maxVBal[a], maxVal ↦ maxVal[a], acc ↦ a])
62      ∧ UNCHANGED ⟨maxVBal, maxVal⟩

64  Phase2a(b) ≜
65    ∧ ¬∃ m ∈ msgs : (m.type = "2a") ∧ (m.bal = b)
66    ∧ ∃ v ∈ Values :
67        ∧ ∃ Q ∈ Quorums :
68            ∃ S ∈ SUBSET {m ∈ msgs : (m.type = "1b") ∧ (m.bal = b)} :
69              ∧ ∀ a ∈ Q : ∃ m ∈ S : m.acc       = a
70              ∧ ∨ ∀ m ∈ S : m.maxVBal = −1
71                ∨ ∃ c ∈ 0 .. (b − 1) :
72                    ∧ ∀ m ∈ S : m.maxVBal ≤ c
73                    ∧ ∃ m ∈ S : ∧ m.maxVBal = c
74                                ∧ m.maxVal = v
75        ∧ Send([type ↦ "2a", bal ↦ b, val ↦ v])
76    ∧ UNCHANGED ⟨maxBal, maxVBal, maxVal⟩

78  Phase2b(a) ≜
79    ∃ m ∈ msgs :
80      ∧ m.type = "2a"
81      ∧ m.bal ≥ maxBal[a]
82      ∧ maxVBal' = [maxVBal EXCEPT ![a] = m.bal]
83      ∧ maxBal' = [maxBal EXCEPT ![a] = m.bal]
84      ∧ maxVal' = [maxVal EXCEPT ![a] = m.val]
85      ∧ Send([type ↦ "2b", bal ↦ m.bal, val ↦ m.val, acc ↦ a])
86  ----------------------------------------------------------------------
87  Next ≜  ∨ ∃ b ∈ Ballots : Phase1a(b) ∨ Phase2a(b)
88          ∨ ∃ a ∈ Acceptors : Phase1b(a) ∨ Phase2b(a)

90  Spec ≜ Init ∧ □[Next]_vars
91  ----------------------------------------------------------------------
92  VotedForIn(a, v, b) ≜ ∃ m ∈ msgs : ∧ m.type = "2b"
93                                     ∧ m.val  = v
94                                     ∧ m.bal  = b
95                                     ∧ m.acc  = a

97  ChosenIn(v, b) ≜ ∃ Q ∈ Quorums :
```

$$\forall\, a \in Q : VotedForIn(a,\, v,\, b)$$

$Chosen(v) \triangleq \exists\, b \in Ballots : ChosenIn(v,\, b)$

$Consistency \triangleq \forall\, v1,\, v2 \in Values : Chosen(v1) \land Chosen(v2) \Rightarrow (v1 = v2)$

$$\begin{aligned}
WontVoteIn(a,\, b) \triangleq\ & \land \forall\, v \in Values : \neg\, VotedForIn(a,\, v,\, b) \\
& \land\ maxBal[a] > b
\end{aligned}$$

$SafeAt(v,\, b) \triangleq$
$\forall\, c \in 0\,..\,(b-1) :$
$\exists\, Q \in Quorums :$
$\forall\, a \in Q : VotedForIn(a,\, v,\, c) \lor WontVoteIn(a,\, c)$

$MsgInv \triangleq$
$\forall\, m \in msgs :$
$\land\, (m.type = \text{“1b”}) \Rightarrow\ \land\ m.bal \leq maxBal[m.acc]$
$\land\ \lor\ \land\ m.maxVal \in Values$
$\land\ m.maxVBal \in Ballots$
conjunct strengthened 2014/04/02 sm
$\land\ VotedForIn(m.acc,\, m.maxVal,\, m.maxVBal)$
$\land\ SafeAt(m.maxVal,\, m.maxVBal)$
$\lor\ \land\ m.maxVal = None$
$\land\ m.maxVBal = -1$
* conjunct added 2014/03/29 sm
$\land \forall\, c \in (m.maxVBal + 1)\,..\,(m.bal - 1) :$
$\neg \exists\, v \in Values : VotedForIn(m.acc,\, v,\, c)$
$\land\, (m.type = \text{“2a”}) \Rightarrow$
$\land\ SafeAt(m.val,\, m.bal)$
$\land \forall\, ma \in msgs : (ma.type = \text{“2a”}) \land (ma.bal = m.bal)$
$\Rightarrow (ma = m)$
$\land\, (m.type = \text{“2b”}) \Rightarrow$
$\land \exists\, ma \in msgs :\ \land\ ma.type = \text{“2a”}$
$\land\ ma.bal = m.bal$
$\land\ ma.val = m.val$
$\land\ m.bal \leq maxVBal[m.acc]$

LEMMA $VotedInv \triangleq$
$MsgInv \land TypeOK \Rightarrow$
$\forall\, a \in Acceptors,\, v \in Values,\, b \in Ballots :$
$VotedForIn(a,\, v,\, b) \Rightarrow SafeAt(v,\, b) \land b \leq maxVBal[a]$
BY DEF $VotedForIn$, $MsgInv$, $Messages$, $TypeOK$

LEMMA $VotedOnce \triangleq$
$MsgInv \Rightarrow\ \forall\, a1,\, a2 \in Acceptors,\, b \in Ballots,\, v1,\, v2 \in Values :$
$VotedForIn(a1,\, v1,\, b) \land VotedForIn(a2,\, v2,\, b) \Rightarrow (v1 = v2)$

BY DEF $MsgInv$, $VotedForIn$

$AccInv \triangleq$
$\forall\, a \in Acceptors :$
$\land\, (maxVal[a] = None) \equiv (maxVBal[a] = -1)$
$\land\, maxVBal[a] \leq maxBal[a]$
conjunct strengthened corresponding to $MsgInv$ 2014/04/02 sm
$\land\, (maxVBal[a] \geq 0) \Rightarrow VotedForIn(a, maxVal[a], maxVBal[a])$ $SafeAt(maxVal[a], maxVBal[a])$
conjunct added corresponding to $MsgInv$ 2014/03/29 sm
$\land\, \forall\, c \in Ballots : c > maxVBal[a] \Rightarrow \neg\exists\, v \in Values : VotedForIn(a, v, c)$

$Inv \triangleq TypeOK \land MsgInv \land AccInv$

The following lemma shows that (the invariant implies that) the predicate $SafeAt(v, b)$ is stable, meaning that once it becomes true, it remains true throughout the rest of the excecution.

LEMMA $SafeAtStable \triangleq Inv \land Next \land TypeOK' \Rightarrow$
$\forall\, v \in Values, b \in Ballots :$
$SafeAt(v, b) \Rightarrow SafeAt(v, b)'$
$\langle 1\rangle$ SUFFICES ASSUME Inv, $Next$, $TypeOK'$,
NEW $v \in Values$, NEW $b \in Ballots$, $SafeAt(v, b)$
PROVE $SafeAt(v, b)'$
OBVIOUS
$\langle 1\rangle$ USE DEF $Send$, Inv, $Ballots$
$\langle 1\rangle$ USE TRUE $\land$ TRUE
$\langle 1\rangle 1$. ASSUME NEW $bb \in Ballots$, $Phase1a(bb)$
PROVE $SafeAt(v, b)'$
BY $\langle 1\rangle 1$, SMT DEF $SafeAt$, $Phase1a$, $VotedForIn$, $WontVoteIn$
$\langle 1\rangle 2$. ASSUME NEW $a \in Acceptors$, $Phase1b(a)$
PROVE $SafeAt(v, b)'$
BY $\langle 1\rangle 2$, $QuorumAssumption$, $SMTT(60)$ DEF $TypeOK$, $SafeAt$, $WontVoteIn$, $VotedForIn$, $Phase1b$
$\langle 1\rangle 3$. ASSUME NEW $bb \in Ballots$, $Phase2a(bb)$
PROVE $SafeAt(v, b)'$
BY $\langle 1\rangle 3$, $QuorumAssumption$, SMT DEF $TypeOK$, $SafeAt$, $WontVoteIn$, $VotedForIn$, $Phase2a$
$\langle 1\rangle 4$. ASSUME NEW $a \in Acceptors$, $Phase2b(a)$
PROVE $SafeAt(v, b)'$
$\langle 2\rangle 1$. PICK $m \in msgs : Phase2b(a)!(m)$
BY $\langle 1\rangle 4$ DEF $Phase2b$
$\langle 2\rangle 2\ \forall\, aa \in Acceptors, bb \in Ballots, vv \in Values :$
$VotedForIn(aa, vv, bb) \Rightarrow VotedForIn(aa, vv, bb)'$
BY $\langle 2\rangle 1$ DEF $TypeOK$, $VotedForIn$
$\langle 2\rangle 3.\ \forall\, aa \in Acceptors, bb \in Ballots : maxBal[aa] > bb \Rightarrow maxBal'[aa] > bb$
BY $\langle 2\rangle 1$ DEF $TypeOK$
$\langle 2\rangle 4$. ASSUME NEW $aa \in Acceptors$, NEW $bb \in Ballots$,
$WontVoteIn(aa, bb)$, NEW $vv \in Values$,
$VotedForIn(aa, vv, bb)'$

PROVE FALSE
⟨3⟩ DEFINE $mm \triangleq [type \mapsto$ "2b", $val \mapsto vv,\ bal \mapsto bb,\ acc \mapsto aa]$
⟨3⟩1. $mm \notin msgs$
BY ⟨2⟩4 DEF *WontVoteIn*, *VotedForIn*
⟨3⟩2. $mm \in msgs'$
⟨4⟩1. PICK $m1 \in msgs'$:
$\land\ m1.type =$ "2b"
$\land\ m1.val = vv$
$\land\ m1.bal = bb$
$\land\ m1.acc = aa$
BY ⟨2⟩4 DEF *VotedForIn*
⟨4⟩.QED BY ⟨4⟩1 DEF *TypeOK*, *Messages* proved by *Zenon*
⟨3⟩3. $aa = a \land m.bal = bb$
BY ⟨2⟩1, ⟨3⟩1, ⟨3⟩2 DEF *TypeOK*
⟨3⟩.QED
BY ⟨2⟩1, ⟨2⟩4, ⟨3⟩3 DEF *Phase2b*, *WontVoteIn*, *TypeOK*
⟨2⟩5 $\forall\, aa \in Acceptors,\ bb \in Ballots : WontVoteIn(aa,\ bb) \Rightarrow WontVoteIn(aa,\ bb)'$
BY ⟨2⟩3, ⟨2⟩4 DEF *WontVoteIn*
⟨2⟩ QED
BY ⟨2⟩2, ⟨2⟩5, *QuorumAssumption* DEF *SafeAt*

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

THEOREM $Invariant \triangleq Spec \Rightarrow \Box Inv$
⟨1⟩ USE DEF *Ballots*
⟨1⟩1. $Init \Rightarrow Inv$
BY DEF *Init*, *Inv*, *TypeOK*, *AccInv*, *MsgInv*, *VotedForIn*

⟨1⟩2. $Inv \land [Next]_{vars} \Rightarrow Inv'$
⟨2⟩ SUFFICES ASSUME *Inv*, *Next*
PROVE Inv'
BY DEF *vars*, *Inv*, *TypeOK*, *MsgInv*, *AccInv*, *SafeAt*, *VotedForIn*, *WontVoteIn*
⟨2⟩ USE DEF *Inv*
⟨2⟩1. $TypeOK'$
⟨3⟩1. ASSUME NEW $b \in Ballots$, $Phase1a(b)$PROVE $TypeOK'$
BY ⟨3⟩1 DEF *TypeOK*, *Phase1a*, *Send*, *Messages*
⟨3⟩2. ASSUME NEW $b \in Ballots$, $Phase2a(b)$PROVE $TypeOK'$
⟨4⟩1. PICK $v \in Values$:
$\land\ Send([type \mapsto$ "2a", $bal \mapsto b,\ val \mapsto v])$
$\land$ UNCHANGED $\langle maxBal,\ maxVBal,\ maxVal\rangle$
BY ⟨3⟩2 DEF *Phase2a*
⟨4⟩.QED
BY ⟨4⟩1 DEF *TypeOK*, *Send*, *Messages*
⟨3⟩3. ASSUME NEW $a \in Acceptors$, $Phase1b(a)$PROVE $TypeOK'$
⟨4⟩.PICK $m \in msgs : Phase1b(a)!(m)$

BY $\langle 3\rangle 3$ DEF $Phase1b$

$\langle 4\rangle$.QED

BY DEF $Send$, $TypeOK$, $Messages$

$\langle 3\rangle 4$. ASSUME NEW $a \in Acceptors$, $Phase2b(a)$PROVE $TypeOK'$

$\langle 4\rangle$.PICK $m \in msgs : Phase2b(a)!(m)$

BY $\langle 3\rangle 4$ DEF $Phase2b$

$\langle 4\rangle$.QED

BY DEF $Send$, $TypeOK$, $Messages$

$\langle 3\rangle$.QED

BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 3\rangle 4$ DEF $Next$

$\langle 2\rangle 2$. $AccInv'$

$\langle 3\rangle 1$. ASSUME NEW $b \in Ballots$, $Phase1a(b)$PROVE $AccInv'$

BY $\langle 2\rangle 1$, $\langle 3\rangle 1$, $SafeAtStable$ DEF $AccInv$, $TypeOK$, $Phase1a$, $VotedForIn$, $Send$

$\langle 3\rangle 2$. ASSUME NEW $b \in Ballots$, $Phase2a(b)$PROVE $AccInv'$

BY $\langle 2\rangle 1$, $\langle 3\rangle 2$, $SafeAtStable$ DEF $AccInv$, $TypeOK$, $Phase2a$, $VotedForIn$, $Send$

$\langle 3\rangle 3$. ASSUME NEW $a \in Acceptors$, $Phase1b(a)$PROVE $AccInv'$

BY $\langle 2\rangle 1$, $\langle 3\rangle 3$, $SafeAtStable$ DEF $AccInv$, $TypeOK$, $Phase1b$, $VotedForIn$, $Send$

$\langle 3\rangle 4$. ASSUME NEW $a \in Acceptors$, $Phase2b(a)$PROVE $AccInv'$

$\langle 4\rangle 1$. PICK $m \in msgs : Phase2b(a)!(m)$

BY $\langle 3\rangle 4$ DEF $Phase2b$

$\langle 4\rangle 2$. $\forall\, acc \in Acceptors :$

$\wedge\, maxVal'[acc] = None \equiv maxVBal'[acc] = -1$

$\wedge\, maxVBal'[acc] \leq maxBal'[acc]$

BY $\langle 2\rangle 1$, $\langle 4\rangle 1$, $NoneNotAValue$ DEF $AccInv$, $TypeOK$, $Messages$

$\langle 4\rangle 3$. $\forall\, aa, vv, bb : VotedForIn(aa, vv, bb)' \equiv$

$VotedForIn(aa, vv, bb) \vee (aa = a \wedge vv = maxVal'[a] \wedge bb = maxVBal'[a])$

BY $\langle 4\rangle 1$, Isa DEF $VotedForIn$, $Send$, $TypeOK$, $Messages$

$\langle 4\rangle 4$. ASSUME NEW $acc \in Acceptors$, $maxVBal'[acc] \geq 0$

PROVE $VotedForIn(acc, maxVal[acc], maxVBal[acc])'$

BY $\langle 4\rangle 1$, $\langle 4\rangle 3$, $\langle 4\rangle 4$ DEF $AccInv$, $TypeOK$

$\langle 4\rangle 5$. ASSUME NEW $acc \in Acceptors$, NEW $c \in Ballots$, $c > maxVBal'[acc]$,

NEW $v \in Values$, $VotedForIn(acc, v, c)'$

PROVE FALSE

BY $\langle 4\rangle 1$, $\langle 4\rangle 3$, $\langle 4\rangle 5$, $\langle 2\rangle 1$ DEF $AccInv$, $TypeOK$

$\langle 4\rangle$.QED BY $\langle 4\rangle 2$, $\langle 4\rangle 4$, $\langle 4\rangle 5$ DEF $AccInv$

$\langle 3\rangle$.QED

BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 3\rangle 4$ DEF $Next$

$\langle 2\rangle 3$. $MsgInv'$

$\langle 3\rangle 1$. ASSUME NEW $b \in Ballots$, $Phase1a(b)$

PROVE $MsgInv'$

$\langle 4\rangle 1$. $\forall\, aa, vv, bb : VotedForIn(aa, vv, bb)' \equiv VotedForIn(aa, vv, bb)$

BY $\langle 3\rangle 1$ DEF $Phase1a$, $Send$, $VotedForIn$

$\langle 4\rangle$.QED

BY $\langle 3\rangle 1$, $\langle 4\rangle 1$, $SafeAtStable$, $\langle 2\rangle 1$ DEF $Phase1a$, $MsgInv$, $TypeOK$, $Messages$, $Send$

$\langle 3\rangle 2$. ASSUME NEW $a \in Acceptors$, $Phase1b(a)$

PROVE $MsgInv'$

$\langle 4\rangle$.PICK $m \in msgs : Phase1b(a)!(m)$
BY $\langle 3\rangle 2$ DEF $Phase1b$

$\langle 4\rangle 1.\ \forall\, aa,\ vv,\ bb : VotedForIn(aa,\ vv,\ bb)' \equiv VotedForIn(aa,\ vv,\ bb)$
BY DEF $Send$, $VotedForIn$

$\langle 4\rangle$.DEFINE $mm \triangleq [type \mapsto$ "1b", $bal \mapsto m.bal,\ maxVBal \mapsto maxVBal[a],$
$maxVal \mapsto maxVal[a],\ acc \mapsto a]$

$\langle 4\rangle 2.\ mm.bal \leq maxBal'[mm.acc]$
BY DEF $TypeOK$, $Messages$

$\langle 4\rangle 3.\ \lor\ \land\, mm.maxVal \in Values$
$\land\, mm.maxVBal \in Ballots$
$\land\, VotedForIn(mm.acc,\ mm.maxVal,\ mm.maxVBal)$
$\lor\ \land\, mm.maxVal = None$
$\land\, mm.maxVBal = -1$
BY DEF $TypeOK$, $AccInv$

$\langle 4\rangle 4.\ \forall\, c \in (mm.maxVBal + 1)\,..\,(mm.bal - 1) :$
$\neg\exists\, v \in Values : VotedForIn(mm.acc,\ v,\ c)$
BY DEF $AccInv$, $TypeOK$, $Messages$

$\langle 4\rangle$.QED
BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$, $\langle 4\rangle 4$, $SafeAtStable$ DEF $MsgInv$, $TypeOK$, $Messages$, $Send$

$\langle 3\rangle 3$. ASSUME NEW $b \in Ballots$, $Phase2a(b)$
PROVE $MsgInv'$

$\langle 4\rangle 1.\ \neg\exists\, m \in msgs : (m.type =$ "2a") $\land\, (m.bal = b)$
BY $\langle 3\rangle 3$ DEF $Phase2a$

$\langle 4\rangle$1a. UNCHANGED $\langle maxBal,\ maxVBal,\ maxVal\rangle$
BY $\langle 3\rangle 3$ DEF $Phase2a$

$\langle 4\rangle 2$. PICK $v \in Values :$
$\land\, \exists\, Q \in Quorums :$
$\exists\, S \in$ SUBSET $\{m \in msgs : (m.type =$ "1b") $\land\, (m.bal = b)\} :$
$\land\, \forall\, a \in Q : \exists\, m \in S : m.acc = a$
$\land\ \lor\, \forall\, m \in S : m.maxVBal = -1$
$\lor\, \exists\, c \in 0\,..\,(b-1) :$
$\land\, \forall\, m \in S : m.maxVBal \leq c$
$\land\, \exists\, m \in S :\ \land\, m.maxVBal = c$
$\land\, m.maxVal = v$
$\land\, Send([type \mapsto$ "2a", $bal \mapsto b,\ val \mapsto v])$
BY $\langle 3\rangle 3$ DEF $Phase2a$

$\langle 4\rangle$.DEFINE $mm \triangleq [type \mapsto$ "2a", $bal \mapsto b,\ val \mapsto v]$

$\langle 4\rangle 3.\ msgs' = msgs \cup \{mm\}$
BY $\langle 4\rangle 2$ DEF $Send$

$\langle 4\rangle 4.\ \forall\, aa,\ vv,\ bb : VotedForIn(aa,\ vv,\ bb)' \equiv VotedForIn(aa,\ vv,\ bb)$
BY $\langle 4\rangle 3$ DEF $VotedForIn$

$\langle 4\rangle 6.\ \forall\, m,\ ma \in msgs' : m.type =$ "2a" $\land\, ma.type =$ "2a" $\land\, ma.bal = m.bal$
$\Rightarrow ma = m$
BY $\langle 4\rangle 1$, $\langle 4\rangle 3$, Isa DEF $MsgInv$

⟨4⟩10. $SafeAt(v,\ b)$
⟨5⟩0. PICK $Q \in Quorums$,
$S \in \text{SUBSET } \{m \in msgs : (m.type = \text{“1b”}) \land (m.bal = b)\} :$
$\land \forall\, a \in Q : \exists\, m \in S : m.acc = a$
$\land \lor \forall\, m \in S : m.maxVBal = -1$
$\lor \exists\, c \in 0 \,..\, (b-1) :$
$\land \forall\, m \in S : m.maxVBal \leq c$
$\land \exists\, m \in S : \land m.maxVBal = c$
$\land m.maxVal = v$
BY ⟨4⟩2, *Zenon*
⟨5⟩1.CASE $\forall\, m \in S : m.maxVBal = -1$
In that case, no acceptor in Q voted in any ballot less than b,
by the last conjunct of *MsgInv* for type “1b” messages, and that's enough
BY ⟨5⟩1, ⟨5⟩0 DEF *TypeOK*, *MsgInv*, *SafeAt*, *WontVoteIn*
⟨5⟩2. ASSUME NEW $c \in 0 \,..\, (b-1)$,
$\forall\, m \in S : m.maxVBal \leq c$,
NEW $ma \in S$, $ma.maxVBal = c$, $ma.maxVal = v$
PROVE $SafeAt(v,\ b)$
⟨6⟩.SUFFICES ASSUME NEW $d \in 0 \,..\, (b-1)$
PROVE $\exists\, QQ \in Quorums : \forall\, q \in QQ :$
$VotedForIn(q,\ v,\ d) \lor WontVoteIn(q,\ d)$
BY DEF *SafeAt*
⟨6⟩1.CASE $d \in 0 \,..\, (c-1)$
The “1b” message for v with *maxVBal* value c must have been safe
according to *MsgInv* for “1b” messages and lemma *VotedInv*,
and that proves the assertion
BY ⟨5⟩2, ⟨6⟩1, *VotedInv* DEF *SafeAt*, *MsgInv*, *TypeOK*, *Messages*
⟨6⟩2.CASE $d = c$
⟨7⟩1. $VotedForIn(ma.acc,\ v,\ c)$
BY ⟨5⟩2 DEF *MsgInv*
⟨7⟩2. $\forall\, q \in Q,\ w \in Values : VotedForIn(q,\ w,\ c) \Rightarrow w = v$
BY ⟨7⟩1, *VotedOnce*, *QuorumAssumption* DEF *TypeOK*, *Messages*
⟨7⟩3. $\forall\, q \in Q : maxBal[q] > c$
BY ⟨5⟩0 DEF *MsgInv*, *TypeOK*, *Messages*
⟨7⟩.QED
BY ⟨6⟩2, ⟨7⟩2, ⟨7⟩3 DEF *WontVoteIn*
⟨6⟩3.CASE $d \in (c+1) \,..\, (b-1)$
By the last conjunct of *MsgInv* for type “1b” messages, no acceptor in Q
voted at any of these ballots.
BY ⟨6⟩3, ⟨5⟩0, ⟨5⟩2 DEF *MsgInv*, *TypeOK*, *Messages*, *WontVoteIn*
⟨6⟩.QED BY ⟨6⟩1, ⟨6⟩2, ⟨6⟩3
⟨5⟩.QED BY ⟨5⟩0, ⟨5⟩1, ⟨5⟩2
⟨4⟩11. $SafeAt(mm.val,\ mm.bal)'$
BY ⟨4⟩10, ⟨2⟩1, *SafeAtStable*
⟨4⟩.QED

This proof used to work.

BY $\langle 2\rangle 1$, $\langle 4\rangle$1a, $\langle 4\rangle 3$, $\langle 4\rangle 4$, $\langle 4\rangle 6$, $\langle 4\rangle 11$, *SafeAtStable*, *Zenon*
DEF *MsgInv*, *TypeOK*, *Messages*

The following decomposition added by *LL* on 21 *Nov* 2014 because *Zenon* failed on this proof. However, *ZenonT*(200) worked.

$\langle 5\rangle$ SUFFICES ASSUME NEW $m \in msgs'$
PROVE $MsgInv!(m)'$
BY DEF *MsgInv*

$\langle 5\rangle 1.\ m.type = \text{“1b”}$
$\Rightarrow\ (\land\ m.bal \leq maxBal[m.acc]$
$\land\ \lor\ \land\ m.maxVal \in Values$
$\land\ m.maxVBal \in Nat$
$\land\ VotedForIn(m.acc,\ m.maxVal,\ m.maxVBal)$
$\lor\ \land\ m.maxVal = None$
$\land\ m.maxVBal = -1$
$\land\ \forall\, c \in m.maxVBal + 1\ldots m.bal - 1 :$
$\neg(\exists\, v\_1 \in Values : VotedForIn(m.acc,\ v\_1,\ c)))'$
BY $\langle 2\rangle 1$, $\langle 4\rangle$1a, $\langle 4\rangle 3$, $\langle 4\rangle 4$, $\langle 4\rangle 6$, $\langle 4\rangle 11$, $SafeAtStable \setminus *$, *Zenon* DEF *MsgInv*, *TypeOK*, *Messages*

$\langle 5\rangle 2.\ m.type = \text{“2a”}$
$\Rightarrow\ (\land\ SafeAt(m.val,\ m.bal)$
$\land\ \forall\, ma \in msgs :$
$ma.type = \text{“2a”} \land ma.bal = m.bal \Rightarrow ma = m)'$
BY $\langle 2\rangle 1$, $\langle 4\rangle$1a, $\langle 4\rangle 3$, $\langle 4\rangle 4$, $\langle 4\rangle 6$, $\langle 4\rangle 11$, $SafeAtStable \setminus *$, *Zenon* DEF *MsgInv*, *TypeOK*, *Messages*

$\langle 5\rangle 3.\ m.type = \text{“2b”}$
$\Rightarrow\ (\land\ \exists\, ma \in msgs :$
$\land\ ma.type = \text{“2a”}$
$\land\ ma.bal = m.bal$
$\land\ ma.val = m.val$
$\land\ m.bal \leq maxVBal[m.acc])'$
BY $\langle 2\rangle 1$, $\langle 4\rangle$1a, $\langle 4\rangle 3$, $\langle 4\rangle 4$, $\langle 4\rangle 6$, $\langle 4\rangle 11$, $SafeAtStable \setminus *$, *Zenon* DEF *MsgInv*, *TypeOK*, *Messages*

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

$\langle 3\rangle 4$. ASSUME NEW $a \in Acceptors$, $Phase2b(a)$
PROVE $MsgInv'$
$\langle 4\rangle$.PICK $m \in msgs : Phase2b(a)!(m)$
BY $\langle 3\rangle 4$ DEF *Phase2b*
$\langle 4\rangle 1.\ \forall\, aa,\ vv,\ bb : VotedForIn(aa,\ vv,\ bb) \Rightarrow VotedForIn(aa,\ vv,\ bb)'$
BY DEF *VotedForIn*, *Send*
$\langle 4\rangle 2.\ \forall\, mm \in msgs : mm.type = \text{“1b”}$
$\Rightarrow\ \forall\, v \in Values,\ c \in (mm.maxVBal + 1)\ldots(mm.bal - 1) :$
$\neg VotedForIn(mm.acc,\ v,\ c) \Rightarrow \neg VotedForIn(mm.acc,\ v,\ c)'$
BY DEF *Send*, *VotedForIn*, *MsgInv*, *TypeOK*, *Messages*
$\langle 4\rangle$.QED

425 BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, *SafeAtStable*, $\langle 2\rangle 1$ DEF *MsgInv*, *Send*, *TypeOK*, *Messages*
426 $\langle 3\rangle 5$. QED
427 BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 3\rangle 4$ DEF *Next*
428 $\langle 2\rangle 4$. QED
429 BY $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$ DEF *Inv*

431 $\langle 1\rangle 3$. QED
432 BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, *PTL* DEF *Spec*

435 THEOREM $Consistent \triangleq Spec \Rightarrow \Box Consistency$
436 $\langle 1\rangle$ USE DEF *Ballots*

438 $\langle 1\rangle 1$. $Inv \Rightarrow Consistency$
439 $\langle 2\rangle$ SUFFICES ASSUME *Inv*,
440 NEW $v1 \in Values$, NEW $v2 \in Values$,
441 NEW $b1 \in Ballots$, NEW $b2 \in Ballots$,
442 $ChosenIn(v1, b1)$, $ChosenIn(v2, b2)$,
443 $b1 \leq b2$
444 PROVE $v1 = v2$
445 BY DEF *Consistency*, *Chosen*
446 $\langle 2\rangle 1$.CASE $b1 = b2$
447 BY $\langle 2\rangle 1$, *VotedOnce*, *QuorumAssumption*, $SMTT(100)$ DEF *ChosenIn*, *Inv*
$\langle 3\rangle 1$. PICK $a1 \in Acceptors : VotedForIn(a1, v1, b1)$
BY *QuorumAssumption* DEF *ChosenIn*
$\langle 3\rangle 2$. PICK $a2 \in Acceptors : VotedForIn(a2, v2, b2)$
BY *QuorumAssumption* DEF *ChosenIn*
$\langle 3\rangle$. QED BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 2\rangle 1$, *VotedOnce* DEF *Inv*
456 $\langle 2\rangle 2$.CASE $b1 < b2$
457 $\langle 3\rangle 1$. $SafeAt(v2, b2)$
458 BY *VotedInv*, *QuorumNonEmpty*, *QuorumAssumption* DEF *ChosenIn*, *Inv*
459 $\langle 3\rangle 2$. PICK $Q2 \in Quorums$:
460 $\forall\, a \in Q2 \quad : VotedForIn(a, v2, b1) \vee WontVoteIn(a, b1)$
461 BY $\langle 3\rangle 1$, $\langle 2\rangle 2$ DEF *SafeAt*
462 $\langle 3\rangle 3$. PICK $Q1 \in Quorums : \forall\, a \in Q1 : VotedForIn(a, v1, b1)$
463 BY DEF *ChosenIn*
464 $\langle 3\rangle 4$. QED
465 BY $\langle 3\rangle 2$, $\langle 3\rangle 3$, *QuorumAssumption*, *VotedOnce*, *Z3* DEF *WontVoteIn*, *Inv*
466 $\langle 2\rangle 3$. QED
467 BY $\langle 2\rangle 1$, $\langle 2\rangle 2$

469 $\langle 1\rangle 2$. QED
470 BY *Invariant*, $\langle 1\rangle 1$, *PTL*

472 ⊢———⊣
473 $chosenBar \triangleq \{v \in Values : Chosen(v)\}$

```
C ≜ INSTANCE Consensus WITH chosen ← chosenBar

THEOREM Refinement ≜ Spec ⇒ C!Spec
⟨1⟩1. Init ⇒ C!Init
  BY QuorumNonEmpty DEF Init, C!Init, chosenBar, Chosen, ChosenIn, VotedForIn

⟨1⟩2. TypeOK′ ∧ Consistency′ ∧ [Next]_vars ⇒ [C!Next]_chosenBar
  ⟨2⟩ SUFFICES ASSUME TypeOK′, Consistency′, Next, chosenBar′ ≠ chosenBar
                PROVE  C!Next
    BY  DEF vars, chosenBar, Chosen, ChosenIn, VotedForIn
  ⟨2⟩1. chosenBar ⊆ chosenBar′
    BY  DEF Send, chosenBar, Chosen, ChosenIn, VotedForIn, Next, Phase1a, Phase1b, Phase2a, Phase2b
  ⟨2⟩2. ∀ v, w ∈ chosenBar′ : v = w
    BY  DEF Consistency, chosenBar, ChosenIn, TypeOK
  ⟨2⟩3. chosenBar = {}
    BY ⟨2⟩1, ⟨2⟩2, SetExtensionality
  ⟨2⟩.QED
    BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3  DEF C!Next, chosenBar

⟨1⟩3. QED
  BY ⟨1⟩1, ⟨1⟩2, Invariant, Consistent, PTL DEF Spec, C!Spec, Inv
```

```
\ * Modification History
\ * Last modified Tue Jul 30 21:29:49 CST 2019 by hengxin
\ * Last modified Mon Jul 22 20:30:39 CST 2019 by hengxin
\ * Last modified Fri Nov 28 10:39:17 PST 2014 by lamport
\ * Last modified Sun Nov 23 14:45:09 PST 2014 by lamport
\ * Last modified Mon Nov 24 02:03:02 CET 2014 by merz
\ * Last modified Sat Nov 22 12:04:19 CET 2014 by merz
\ * Last modified Fri Nov 21 17:40:41 PST 2014 by lamport
\ * Last modified Tue Mar 18 11:37:57 CET 2014 by doligez
\ * Last modified Sat Nov 24 18:53:09 GMT − 03:00 2012 by merz
\ * Created Sat Nov 17 16:02:06 PST 2012 by lamport
```