```
┌──────────────────────────── MODULE Paxos ────────────────────────────┐
 Specification and Verification of Basic Paxos.

 See http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html ≠ paxos-simple
EXTENDS Integers, TLAPS, TLC
├──────────────────────────────────────────────────────────────────────┤
CONSTANTS Acceptors, Values, Quorums

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

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

Ballots ≜ Nat

None ≜ CHOOSE v : v ∉ Values

LEMMA NoneNotAValue ≜ None ∉ Values
BY NoSetContainsEverything DEF None

Messages ≜      [type : {"1a"}, bal : Ballots]
           ∪    [type : {"1b"}, bal : Ballots, maxVBal : Ballots ∪ { − 1},
                   maxVal : Values ∪ {None}, acc : Acceptors]
           ∪    [type : {"2a"}, bal : Ballots, val : Values]
           ∪    [type : {"2b"}, bal : Ballots, val : Values, acc : Acceptors]
├──────────────────────────────────────────────────────────────────────┤
VARIABLES msgs,        the set of messages that have been sent.
          maxBal,      maxBal[a]: the highest-number ballot acceptor a has participated in.
          maxVBal,     maxVBal[a]: the highest ballot in which a has voted;
          maxVal       maxVal[a]: the value it voted for in that ballot.

vars ≜ ⟨msgs, maxBal, maxVBal, maxVal⟩

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

Send(m) ≜ msgs′ = msgs ∪ {m}
├──────────────────────────────────────────────────────────────────────┤
Init ≜ ∧ msgs = {}
       ∧ maxVBal = [a ∈ Acceptors ↦ − 1]
       ∧ maxBal  = [a ∈ Acceptors ↦ − 1]
       ∧ maxVal  = [a ∈ Acceptors ↦ None]

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

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

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

Phase2a(b) ≜
  ∧ ¬∃ m ∈ msgs : (m.type = "2a") ∧ (m.bal = b)
  ∧ ∃ v ∈ Values :
       ∧ ∃ Q ∈ Quorums :
            ∃ S ∈ SUBSET {m ∈ msgs : (m.type = "1b") ∧ (m.bal = b)} :
               ∧ ∀ a ∈ Q : ∃ m ∈ S : m.acc     = a
               ∧ ∨ ∀ m ∈ S : m.maxVBal = −1
                 ∨ ∃ c ∈ 0 . . (b − 1) :
                      ∧ ∀ m ∈ S : m.maxVBal ≤ c
                      ∧ ∃ m ∈ S : ∧ m.maxVBal = c
                                  ∧ m.maxVal = v
       ∧ Send([type ↦ "2a", bal ↦ b, val ↦ v])
  ∧ UNCHANGED ⟨maxBal, maxVBal, maxVal⟩

Phase2b(a) ≜
  ∃ m ∈ msgs :
     ∧ m.type = "2a"
     ∧ m.bal ≥ maxBal[a]
     ∧ Send([type ↦ "2b", bal ↦ m.bal, val ↦ m.val, acc ↦ a])
     ∧ maxVBal' = [maxVBal EXCEPT ![a] = m.bal]
     ∧ maxBal' = [maxBal EXCEPT ![a] = m.bal]
     ∧ maxVal' = [maxVal EXCEPT ![a] = m.val]
--------------------------------------------------------------------------
Next ≜ ∨ ∃ b ∈ Ballots : Phase1a(b) ∨ Phase2a(b)
       ∨ ∃ a ∈ Acceptors : Phase1b(a) ∨ Phase2b(a)

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

ChosenIn(v, b) ≜ ∃ Q ∈ Quorums :
```

```
                        $\forall a \in Q : \mathit{VotedForIn}(a, v, b)$

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

$\mathit{Consistency} \triangleq \forall v1, v2 \in \mathit{Values} : \mathit{Chosen}(v1) \land \mathit{Chosen}(v2) \Rightarrow (v1 = v2)$
--------------------------------------------------------------------------------
$\mathit{WontVoteIn}(a, b) \triangleq \land \forall v \in \mathit{Values} : \neg \mathit{VotedForIn}(a, v, b)$
                          $\land\ \mathit{maxBal}[a] > b$

$\mathit{SafeAt}(v, b) \triangleq$
  $\forall c \in 0 \,..\, (b - 1) :$
    $\exists Q \in \mathit{Quorums} :$
      $\forall a \in Q : \mathit{VotedForIn}(a, v, c) \lor \mathit{WontVoteIn}(a, c)$
--------------------------------------------------------------------------------
$\mathit{MsgInv} \triangleq$
  $\forall m \in \mathit{msgs} :$
     $\land (m.\mathit{type} = \text{“1b”}) \Rightarrow \land m.\mathit{bal} \leq \mathit{maxBal}[m.\mathit{acc}]$
                              $\land \lor \land m.\mathit{maxVal} \in \mathit{Values}$
                                   $\land m.\mathit{maxVBal} \in \mathit{Ballots}$
                                    conjunct strengthened 2014/04/02 sm
                                   $\land \mathit{VotedForIn}(m.\mathit{acc}, m.\mathit{maxVal}, m.\mathit{maxVBal})$
                        $\land \mathit{SafeAt}(m.\mathit{maxVal}, m.\mathit{maxVBal})$
                                $\lor \land m.\mathit{maxVal} = \mathit{None}$
                                  $\land m.\mathit{maxVBal} = -1$
                              * conjunct added 2014/03/29 sm
                            $\land \forall c \in (m.\mathit{maxVBal} + 1) \,..\, (m.\mathit{bal} - 1) :$
                                $\neg \exists v \in \mathit{Values} : \mathit{VotedForIn}(m.\mathit{acc}, v, c)$
     $\land (m.\mathit{type} = \text{“2a”}) \Rightarrow$
          $\land \mathit{SafeAt}(m.\mathit{val}, m.\mathit{bal})$
          $\land \forall ma \in \mathit{msgs} : (ma.\mathit{type} = \text{“2a”}) \land (ma.\mathit{bal} = m.\mathit{bal})$
                                   $\Rightarrow (ma = m)$
     $\land (m.\mathit{type} = \text{“2b”}) \Rightarrow$
          $\land \exists ma \in \mathit{msgs} : \land ma.\mathit{type} = \text{“2a”}$
                              $\land ma.\mathit{bal} = m.\mathit{bal}$
                              $\land ma.\mathit{val} = m.\mathit{val}$
          $\land m.\mathit{bal} \leq \mathit{maxVBal}[m.\mathit{acc}]$
--------------------------------------------------------------------------------
LEMMA $\mathit{VotedInv} \triangleq$
        $\mathit{MsgInv} \land \mathit{TypeOK} \Rightarrow$
          $\forall a \in \mathit{Acceptors}, v \in \mathit{Values}, b \in \mathit{Ballots} :$
            $\mathit{VotedForIn}(a, v, b) \Rightarrow \mathit{SafeAt}(v, b) \land b \leq \mathit{maxVBal}[a]$
BY DEF $\mathit{VotedForIn}, \mathit{MsgInv}, \mathit{Messages}, \mathit{TypeOK}$

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

```
BY  DEF MsgInv, VotedForIn

AccInv ≜
  ∀ a ∈ Acceptors :
     ∧ (maxVal[a]  = None) ≡ (maxVBal[a] = −1)
     ∧ maxVBal[a] ≤ maxBal[a]
      conjunct strengthened corresponding to MsgInv 2014/04/02 sm
     ∧ (maxVBal[a] ≥ 0) ⇒ VotedForIn(a, maxVal[a], maxVBal[a])   SafeAt(maxVal[a], maxVBal[a])
      conjunct added corresponding to MsgInv 2014/03/29 sm
     ∧ ∀ c ∈ Ballots : c > maxVBal[a] ⇒ ¬∃ v ∈ Values : VotedForIn(a, v, c)
├──────────────────────────────────────────────────────────────────┤
Inv ≜ TypeOK ∧ MsgInv ∧ 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 ≜ Inv ∧ Next ∧ TypeOK′ ⇒
                                ∀ v ∈ Values, b ∈ Ballots :
                                        SafeAt(v, b) ⇒ SafeAt(v, b)′
⟨1⟩ SUFFICES ASSUME Inv, Next, TypeOK′,
                    NEW v ∈ Values, NEW b ∈ Ballots, SafeAt(v, b)
             PROVE  SafeAt(v, b)′
  OBVIOUS
⟨1⟩ USE  DEF Send, Inv, Ballots
⟨1⟩ USE TRUE ∧ TRUE
⟨1⟩1. ASSUME NEW bb ∈ Ballots, Phase1a(bb)
      PROVE  SafeAt(v, b)′
  BY ⟨1⟩1, SMT DEF SafeAt, Phase1a, VotedForIn, WontVoteIn
⟨1⟩2. ASSUME NEW a ∈ Acceptors, Phase1b(a)
      PROVE  SafeAt(v, b)′
  BY ⟨1⟩2, QuorumAssumption, SMTT(60) DEF TypeOK, SafeAt, WontVoteIn, VotedForIn, Phase1b
⟨1⟩3. ASSUME NEW bb ∈ Ballots, Phase2a(bb)
      PROVE  SafeAt(v, b)′
  BY ⟨1⟩3, QuorumAssumption, SMT DEF TypeOK, SafeAt, WontVoteIn, VotedForIn, Phase2a
⟨1⟩4. ASSUME NEW a ∈ Acceptors, Phase2b(a)
      PROVE  SafeAt(v, b)′
  ⟨2⟩1. PICK m ∈ msgs : Phase2b(a)!(m)
    BY ⟨1⟩4  DEF Phase2b
  ⟨2⟩2 ∀ aa ∈ Acceptors, bb ∈ Ballots, vv ∈ Values :
           VotedForIn(aa, vv, bb) ⇒ VotedForIn(aa, vv, bb)′
    BY ⟨2⟩1  DEF TypeOK, VotedForIn
  ⟨2⟩3. ∀ aa ∈ Acceptors, bb ∈ Ballots : maxBal[aa] > bb ⇒ maxBal′[aa] > bb
    BY ⟨2⟩1  DEF TypeOK
  ⟨2⟩4. ASSUME NEW aa ∈ Acceptors, NEW bb ∈ Ballots,
               WontVoteIn(aa, bb), NEW vv ∈ Values,
               VotedForIn(aa, vv, bb)′
```

```
          PROVE FALSE
      ⟨3⟩ DEFINE mm ≜ [type ↦ "2b", val ↦ vv, bal ↦ bb, acc ↦ aa]
      ⟨3⟩1. mm ∉ msgs
        BY ⟨2⟩4 DEF WontVoteIn, VotedForIn
      ⟨3⟩2. mm ∈ msgs′
        ⟨4⟩1. PICK m1 ∈ msgs′ :
                   ∧ m1.type = "2b"
                   ∧ m1.val = vv
                   ∧ m1.bal = bb
                   ∧ m1.acc = aa
          BY ⟨2⟩4 DEF VotedForIn
        ⟨4⟩.QED BY ⟨4⟩1 DEF TypeOK, Messages  proved by Zenon
      ⟨3⟩3. aa = a ∧ 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 ∀ aa ∈ Acceptors, bb ∈ Ballots : WontVoteIn(aa, bb) ⇒ 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 ≜ Spec ⇒ □Inv
⟨1⟩ USE DEF Ballots
⟨1⟩1. Init ⇒ Inv
  BY DEF Init, Inv, TypeOK, AccInv, MsgInv, VotedForIn

⟨1⟩2. Inv ∧ [Next]_vars ⇒ 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 ∈ Ballots, Phase1a(b)PROVE TypeOK′
      BY ⟨3⟩1 DEF TypeOK, Phase1a, Send, Messages
    ⟨3⟩2. ASSUME NEW b ∈ Ballots, Phase2a(b)PROVE TypeOK′
      ⟨4⟩1. PICK v ∈ Values :
                 ∧ Send([type ↦ "2a", bal ↦ b, val ↦ v])
                 ∧ UNCHANGED ⟨maxBal, maxVBal, maxVal⟩
        BY ⟨3⟩2 DEF Phase2a
      ⟨4⟩.QED
        BY ⟨4⟩1 DEF TypeOK, Send, Messages
    ⟨3⟩3. ASSUME NEW a ∈ Acceptors, Phase1b(a)PROVE TypeOK′
      ⟨4⟩.PICK m ∈ msgs : Phase1b(a)!(m)
      BY ⟨3⟩3  DEF Phase1b
    ⟨4⟩.QED
      BY  DEF Send, TypeOK, Messages
  ⟨3⟩4. ASSUME NEW a ∈ Acceptors, Phase2b(a)PROVE  TypeOK′
    ⟨4⟩.PICK m ∈ msgs : Phase2b(a)!(m)
      BY ⟨3⟩4  DEF Phase2b
    ⟨4⟩.QED
      BY  DEF Send, TypeOK, Messages
  ⟨3⟩.QED
    BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, ⟨3⟩4  DEF Next
⟨2⟩2. AccInv′
  ⟨3⟩1. ASSUME NEW b ∈ Ballots, Phase1a(b)PROVE  AccInv′
    BY ⟨2⟩1, ⟨3⟩1, SafeAtStable DEF AccInv, TypeOK, Phase1a, VotedForIn, Send
  ⟨3⟩2. ASSUME NEW b ∈ Ballots, Phase2a(b)PROVE  AccInv′
      BY ⟨2⟩1, ⟨3⟩2, SafeAtStable DEF AccInv, TypeOK, Phase2a, VotedForIn, Send
  ⟨3⟩3. ASSUME NEW a ∈ Acceptors, Phase1b(a)PROVE  AccInv′
      BY ⟨2⟩1, ⟨3⟩3, SafeAtStable DEF AccInv, TypeOK, Phase1b, VotedForIn, Send
  ⟨3⟩4. ASSUME NEW a ∈ Acceptors, Phase2b(a)PROVE  AccInv′
    ⟨4⟩1. PICK m ∈ msgs : Phase2b(a)!(m)
      BY ⟨3⟩4  DEF Phase2b
    ⟨4⟩2. ∀ acc ∈ Acceptors :
            ∧ maxVal′[acc] = None ≡ maxVBal′[acc] =  −1
            ∧ maxVBal′[acc] ≤ maxBal′[acc]
      BY ⟨2⟩1, ⟨4⟩1, NoneNotAValue DEF AccInv, TypeOK, Messages
    ⟨4⟩3. ∀ aa, vv, bb : VotedForIn(aa, vv, bb)′ ≡
                         VotedForIn(aa, vv, bb) ∨ (aa = a ∧ vv = maxVal′[a] ∧ bb = maxVBal′[a])
      BY ⟨4⟩1, Isa DEF VotedForIn, Send, TypeOK, Messages
    ⟨4⟩4. ASSUME NEW acc ∈ Acceptors, maxVBal′[acc] ≥ 0
          PROVE   VotedForIn(acc, maxVal[acc], maxVBal[acc])′
      BY ⟨4⟩1, ⟨4⟩3, ⟨4⟩4  DEF AccInv, TypeOK
    ⟨4⟩5. ASSUME NEW acc ∈ Acceptors, NEW c ∈ Ballots, c > maxVBal′[acc],
                 NEW v ∈ Values, VotedForIn(acc, v, c)′
          PROVE  FALSE
      BY ⟨4⟩1, ⟨4⟩3, ⟨4⟩5, ⟨2⟩1  DEF AccInv, TypeOK
    ⟨4⟩.QED  BY ⟨4⟩2, ⟨4⟩4, ⟨4⟩5  DEF AccInv
  ⟨3⟩.QED
    BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, ⟨3⟩4  DEF Next
⟨2⟩3. MsgInv′
  ⟨3⟩1. ASSUME NEW b ∈ Ballots, Phase1a(b)
        PROVE  MsgInv′
    ⟨4⟩1. ∀ aa, vv, bb : VotedForIn(aa, vv, bb)′ ≡ VotedForIn(aa, vv, bb)
      BY ⟨3⟩1  DEF Phase1a, Send, VotedForIn
    ⟨4⟩.QED
      BY ⟨3⟩1, ⟨4⟩1, SafeAtStable, ⟨2⟩1  DEF Phase1a, MsgInv, TypeOK, Messages, Send
  ⟨3⟩2. ASSUME NEW a ∈ Acceptors, Phase1b(a)
```

```
        PROVE  MsgInv′
    ⟨4⟩.PICK m ∈ msgs : Phase1b(a)!(m)
      BY ⟨3⟩2 DEF Phase1b
    ⟨4⟩1. ∀ aa, vv, bb : VotedForIn(aa, vv, bb)′ ≡ VotedForIn(aa, vv, bb)
      BY DEF Send, VotedForIn
    ⟨4⟩.DEFINE mm ≜ [type ↦ "1b", bal ↦ m.bal, maxVBal ↦ maxVBal[a],
                     maxVal ↦ maxVal[a], acc ↦ a]
    ⟨4⟩2. mm.bal ≤ maxBal′[mm.acc]
      BY DEF TypeOK, Messages
    ⟨4⟩3. ∨ ∧ mm.maxVal ∈ Values
            ∧ mm.maxVBal ∈ Ballots
            ∧ VotedForIn(mm.acc, mm.maxVal, mm.maxVBal)
          ∨ ∧ mm.maxVal = None
            ∧ mm.maxVBal = −1
      BY DEF TypeOK, AccInv
    ⟨4⟩4. ∀ c ∈ (mm.maxVBal + 1) . . (mm.bal − 1) :
            ¬∃ v ∈ Values : VotedForIn(mm.acc, v, c)
      BY DEF AccInv, TypeOK, Messages
    ⟨4⟩.QED
      BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3, ⟨4⟩4, SafeAtStable DEF MsgInv, TypeOK, Messages, Send
  ⟨3⟩3. ASSUME NEW b ∈ Ballots, Phase2a(b)
        PROVE  MsgInv′
    ⟨4⟩1. ¬∃ m ∈ msgs : (m.type = "2a") ∧ (m.bal = b)
      BY ⟨3⟩3 DEF Phase2a
    ⟨4⟩1a. UNCHANGED ⟨maxBal, maxVBal, maxVal⟩
      BY ⟨3⟩3 DEF Phase2a
    ⟨4⟩2. PICK v ∈ Values :
             ∧ ∃ Q ∈ Quorums :
                  ∃ S ∈ SUBSET {m ∈ msgs : (m.type = "1b") ∧ (m.bal = b)} :
                    ∧ ∀ a ∈ Q : ∃ m ∈ S : m.acc      = a
                    ∧ ∨ ∀ m ∈ S : m.maxVBal = −1
                      ∨ ∃ c  ∈ 0 . . (b − 1) :
                          ∧ ∀ m ∈ S : m.maxVBal ≤ c
                          ∧ ∃ m ∈ S : ∧ m.maxVBal = c
                                      ∧ m.maxVal = v
             ∧ Send([type ↦ "2a", bal ↦ b, val ↦ v])
      BY ⟨3⟩3 DEF Phase2a
    ⟨4⟩.DEFINE mm ≜ [type ↦ "2a", bal ↦ b, val ↦ v]
    ⟨4⟩3. msgs′ = msgs ∪ {mm}
      BY ⟨4⟩2 DEF Send
    ⟨4⟩4. ∀ aa, vv, bb : VotedForIn(aa, vv, bb)′ ≡ VotedForIn(aa, vv, bb)
      BY ⟨4⟩3 DEF VotedForIn
    ⟨4⟩6. ∀ m, ma ∈ msgs′ : m.type = "2a" ∧ ma.type = "2a" ∧ ma.bal = m.bal
                             ⇒ ma = m
      BY ⟨4⟩1, ⟨4⟩3, Isa DEF MsgInv
```

```
⟨4⟩10. $SafeAt(v, b)$
  ⟨5⟩0. PICK $Q \in Quorums$,
                 $S \in$ 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 $\langle 2\rangle 1, \langle 4\rangle 1a, \langle 4\rangle 3, \langle 4\rangle 4, \langle 4\rangle 6, \langle 4\rangle 11, \mathit{SafeAtStable}, \mathit{Zenon}$
         DEF $\mathit{MsgInv}, \mathit{TypeOK}, \mathit{Messages}$
The following decomposition added by $LL$ on 21 $Nov$ 2014 because
$\mathit{Zenon}$ failed on this proof. However, $\mathit{ZenonT}(200)$ worked.
   $\langle 5\rangle$ SUFFICES ASSUME NEW $m \in msgs'$
            PROVE $\mathit{MsgInv}!(m)'$
    BY DEF $\mathit{MsgInv}$
   $\langle 5\rangle 1.$ $m.type = $ "1b"
        $\Rightarrow (\land m.bal \leq maxBal[m.acc]$
           $\land \lor \land m.maxVal \in Values$
                $\land m.maxVBal \in Nat$
                $\land VotedForIn(m.acc, m.maxVal, m.maxVBal)$
              $\lor \land m.maxVal = None$
                $\land m.maxVBal = -1$
           $\land \forall c \in m.maxVBal + 1 \,..\, m.bal - 1 :$
                $\neg(\exists v\_1 \in Values : VotedForIn(m.acc, v\_1, c)))'$
    BY $\langle 2\rangle 1, \langle 4\rangle 1a, \langle 4\rangle 3, \langle 4\rangle 4, \langle 4\rangle 6, \langle 4\rangle 11, \mathit{SafeAtStable} \setminus *, \mathit{Zenon}$ DEF $\mathit{MsgInv}, \mathit{TypeOK},$
      $\mathit{Messages}$
   $\langle 5\rangle 2.$ $m.type = $ "2a"
        $\Rightarrow (\land SafeAt(m.val, m.bal)$
           $\land \forall ma \in msgs :$
                $ma.type = $ "2a" $\land ma.bal = m.bal \Rightarrow ma = m)'$
    BY $\langle 2\rangle 1, \langle 4\rangle 1a, \langle 4\rangle 3, \langle 4\rangle 4, \langle 4\rangle 6, \langle 4\rangle 11, \mathit{SafeAtStable} \setminus *, \mathit{Zenon}$ DEF $\mathit{MsgInv}, \mathit{TypeOK},$
      $\mathit{Messages}$
   $\langle 5\rangle 3.$ $m.type = $ "2b"
        $\Rightarrow (\land \exists ma \in msgs :$
                $\land ma.type = $ "2a"
                $\land ma.bal = m.bal$
                $\land ma.val = m.val$
           $\land m.bal \leq maxVBal[m.acc])'$
    BY $\langle 2\rangle 1, \langle 4\rangle 1a, \langle 4\rangle 3, \langle 4\rangle 4, \langle 4\rangle 6, \langle 4\rangle 11, \mathit{SafeAtStable} \setminus *$ , $\mathit{Zenon}$ DEF $\mathit{MsgInv}, \mathit{TypeOK},$
      $\mathit{Messages}$
   $\langle 5\rangle 4.$ 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 $\mathit{MsgInv}'$
   $\langle 4\rangle$.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 = $ "1b"
               $\Rightarrow \forall v \in Values, c \in (mm.maxVBal + 1) \,..\, (mm.bal - 1) :$
                     $\neg VotedForIn(mm.acc, v, c) \Rightarrow \neg VotedForIn(mm.acc, v, c)'$
     BY DEF $Send, VotedForIn, MsgInv, TypeOK, Messages$
   $\langle 4\rangle$.QED
```

```
          BY ⟨4⟩1, ⟨4⟩2, SafeAtStable, ⟨2⟩1  DEF MsgInv, Send, TypeOK, Messages
      ⟨3⟩5. QED
        BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, ⟨3⟩4  DEF Next
  ⟨2⟩4. QED
    BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3  DEF Inv

⟨1⟩3. QED
  BY ⟨1⟩1, ⟨1⟩2, PTL DEF Spec


THEOREM Consistent ≜ Spec ⇒ □Consistency
⟨1⟩ USE  DEF Ballots

⟨1⟩1. Inv ⇒ Consistency
  ⟨2⟩ SUFFICES ASSUME Inv,
                      NEW v1 ∈ Values,  NEW v2 ∈ Values,
                      NEW b1 ∈ Ballots, NEW b2 ∈ Ballots,
                      ChosenIn(v1, b1), ChosenIn(v2, b2),
                      b1 ≤ b2
               PROVE  v1 = v2
    BY  DEF Consistency, Chosen
  ⟨2⟩1.CASE b1 = b2
    BY ⟨2⟩1, VotedOnce, QuorumAssumption, SMTT(100) DEF ChosenIn, Inv
  ⟨3⟩1. PICK  a1 ∈ Acceptors : VotedForIn(a1, v1, b1)
   BY  QuorumAssumption  DEF  ChosenIn
  ⟨3⟩2. PICK  a2 ∈ Acceptors : VotedForIn(a2, v2, b2)
   BY  QuorumAssumption  DEF  ChosenIn
  ⟨3⟩. QED  BY  ⟨3⟩1, ⟨3⟩2, ⟨2⟩1, VotedOnce  DEF  Inv
  ⟨2⟩2.CASE b1 < b2
    ⟨3⟩1. SafeAt(v2, b2)
      BY VotedInv, QuorumNonEmpty, QuorumAssumption DEF ChosenIn, Inv
    ⟨3⟩2. PICK Q2 ∈ Quorums :
                 ∀ a ∈ Q2     : VotedForIn(a, v2, b1) ∨ WontVoteIn(a, b1)
      BY ⟨3⟩1, ⟨2⟩2  DEF SafeAt
    ⟨3⟩3. PICK Q1 ∈ Quorums : ∀ a ∈ Q1 : VotedForIn(a, v1, b1)
      BY  DEF ChosenIn
    ⟨3⟩4. QED
      BY ⟨3⟩2, ⟨3⟩3, QuorumAssumption, VotedOnce, Z3 DEF WontVoteIn, Inv
  ⟨2⟩3. QED
    BY ⟨2⟩1, ⟨2⟩2

⟨1⟩2. QED
  BY Invariant, ⟨1⟩1, PTL

├──────────────────────────────────────────────────────────────────────┤
chosenBar ≜ {v ∈ 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 Mon Jul 22 20:58:32 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
```