```
┌──────────────────────────── MODULE Voting ────────────────────────────┐
EXTENDS Sets
├───────────────────────────────────────────────────────────────────────┤
CONSTANT Value, Acceptor, Quorum

ASSUME QuorumAssumption ≜
    ∧  ∀ Q ∈ Quorum : Q ⊆ Acceptor
    ∧  ∀ Q1, Q2 ∈ Quorum : Q1 ∩ Q2 ≠ {}

THEOREM QuorumNonEmpty ≜ ∀ Q ∈ Quorum : Q ≠ {}
BY QuorumAssumption

Ballot ≜ Nat
├───────────────────────────────────────────────────────────────────────┤
VARIABLES votes, maxBal

TypeOK ≜ ∧ votes ∈ [Acceptor → SUBSET (Ballot × Value)]
         ∧ maxBal ∈ [Acceptor → Ballot ∪ {−1}]
├───────────────────────────────────────────────────────────────────────┤
VotedFor(a, b, v) ≜ ⟨b, v⟩ ∈ votes[a]

DidNotVoteAt(a, b) ≜ ∀ v ∈ Value : ¬VotedFor(a, b, v)

ShowsSafeAt(Q, b, v) ≜
  ∧ ∀ a ∈ Q : maxBal[a] ≥ b  have promised
  ∧ ∃ c ∈ −1 .. (b − 1) :
      ∧ (c ≠ −1) ⇒ ∃ a ∈ Q : VotedFor(a, c, v)
      ∧ ∀ d ∈ (c + 1) .. (b − 1), a ∈ Q : DidNotVoteAt(a, d)
├───────────────────────────────────────────────────────────────────────┤
Init ≜
    ∧ votes = [a ∈ Acceptor ↦ {}]
    ∧ maxBal = [a ∈ Acceptor ↦ −1]

IncreaseMaxBal(a, b) ≜
  ∧ b > maxBal[a]
  ∧ maxBal′ = [maxBal EXCEPT ![a] = b]  make promise
  ∧ UNCHANGED votes

VoteFor(a, b, v) ≜
    ∧  maxBal[a] ≤ b  keep promise
    ∧  ∀ vt ∈ votes[a] : vt[1] ≠ b
    ∧  ∀ c ∈ Acceptor \ {a} :
          ∀ vt ∈ votes[c] : (vt[1] = b) ⇒ (vt[2] = v)
    ∧  ∃ Q ∈ Quorum : ShowsSafeAt(Q, b, v)  safe to vote
    ∧  votes′ = [votes EXCEPT ![a] = votes[a] ∪ {⟨b, v⟩}]  vote
    ∧  maxBal′ = [maxBal EXCEPT ![a] = b]  make promise
├───────────────────────────────────────────────────────────────────────┤
```

```
Next ≜
    ∃ a ∈ Acceptor, b ∈ Ballot :
        ∨ IncreaseMaxBal(a, b)
        ∨ ∃ v ∈ Value : VoteFor(a, b, v)

Spec ≜ Init ∧ □[Next]_⟨votes, maxBal⟩
├──────────────────────────────────────────────────────────────┤
ChosenAt(b, v) ≜
    ∃ Q ∈ Quorum : ∀ a ∈ Q : VotedFor(a, b, v)

chosen ≜ {v ∈ Value : ∃ b ∈ Ballot : ChosenAt(b, v)}
├──────────────────────────────────────────────────────────────┤
CannotVoteAt(a, b) ≜
     ∧ maxBal[a] > b
     ∧ DidNotVoteAt(a, b)

NoneOtherChoosableAt(b, v) ≜
    ∃ Q ∈ Quorum :
       ∀ a ∈ Q : VotedFor(a, b, v) ∨ CannotVoteAt(a, b)

SafeAt(b, v) ≜
    ∀ c ∈ 0 . . (b − 1) : NoneOtherChoosableAt(c, v)

VotesSafe ≜
    ∀ a ∈ Acceptor, b ∈ Ballot, v ∈ Value :
        VotedFor(a, b, v) ⇒ SafeAt(b, v)

OneVote ≜
    ∀ a ∈ Acceptor, b ∈ Ballot, v, w ∈ Value :
        VotedFor(a, b, v) ∧ VotedFor(a, b, w) ⇒ (v = w)

OneValuePerBallot ≜
    ∀ a1, a2 ∈ Acceptor, b ∈ Ballot, v1, v2 ∈ Value :
        VotedFor(a1, b, v1) ∧ VotedFor(a2, b, v2) ⇒ (v1 = v2)

Inv ≜ TypeOK ∧ VotesSafe ∧ OneValuePerBallot
├──────────────────────────────────────────────────────────────┤
THEOREM AllSafeAtZero ≜ ∀ v ∈ Value : SafeAt(0, v)
  BY DEF SafeAt

THEOREM ChoosableThm ≜
                  ∀ b ∈ Ballot, v ∈ Value :
                     ChosenAt(b, v) ⇒ NoneOtherChoosableAt(b, v)
  BY DEF ChosenAt, NoneOtherChoosableAt

THEOREM OneVoteThm ≜ OneValuePerBallot ⇒ OneVote
  BY DEF OneValuePerBallot, OneVote
├──────────────────────────────────────────────────────────────┤
```

```
THEOREM VotesSafeImpliesConsistency ≜
   ASSUME VotesSafe, OneVote, chosen ≠ {}
   PROVE  ∃ v ∈ Value : chosen = {v}
⟨1⟩1. PICK v ∈ Value : v ∈ chosen
  BY DEF chosen
⟨1⟩2. SUFFICES ASSUME NEW w ∈ chosen
               PROVE  w = v
  BY ⟨1⟩1, ⟨1⟩2
⟨1⟩3. ASSUME NEW b1 ∈ Ballot, NEW b2 ∈ Ballot, b1 < b2,
             NEW v1 ∈ Value, NEW v2 ∈ Value,
             ChosenAt(b1, v1) ∧ ChosenAt(b2, v2)
      PROVE  v1 = v2
  ⟨2⟩1. SafeAt(b2, v2)
    BY ⟨1⟩3, QuorumAssumption, SMT DEF ChosenAt, VotesSafe
  ⟨2⟩2. QED
    BY ⟨1⟩3, ⟨2⟩1, QuorumAssumption, Z3
    DEFS CannotVoteAt, DidNotVoteAt, OneVote,
         ChosenAt, NoneOtherChoosableAt, Ballot, SafeAt
⟨1⟩4. QED
  BY QuorumAssumption, ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, Z3
  DEFS Ballot, ChosenAt, OneVote, chosen

THEOREM ShowsSafety ≜
            TypeOK ∧ VotesSafe ∧ OneValuePerBallot ⇒
              ∀ Q ∈ Quorum, b ∈ Ballot, v ∈ Value :
                ShowsSafeAt(Q, b, v) ⇒ SafeAt(b, v)
  BY QuorumAssumption, Z3
  DEFS Ballot, TypeOK, VotesSafe, OneValuePerBallot, SafeAt,
    ShowsSafeAt, CannotVoteAt, NoneOtherChoosableAt, DidNotVoteAt

THEOREM SafeAtStable ≜ Inv ∧ Next ∧ TypeOK' ⇒
                          ∀ b ∈ Ballot, v ∈ Value :
                            SafeAt(b, v) ⇒ SafeAt(b, v)'
  OMITTED
-----------------------------------------------------------------------
THEOREM Invariance ≜ Spec ⇒ □Inv
⟨1⟩ USE DEF Inv
⟨1⟩1. Init ⇒ Inv
  BY DEF Init, TypeOK, VotesSafe, OneValuePerBallot, VotedFor
⟨1⟩2. Inv ∧ [Next]_⟨votes, maxBal⟩ ⇒ Inv'
  ⟨2⟩ SUFFICES ASSUME Inv, [Next]_⟨votes, maxBal⟩
               PROVE  Inv'
    OBVIOUS
  ⟨2⟩1.CASE Next
    ⟨3⟩ SUFFICES ASSUME NEW a ∈ Acceptor, NEW b ∈ Ballot,
                          ∨ IncreaseMaxBal(a, b)
                          ∨ ∃ v ∈ Value : VoteFor(a, b, v)
                    PROVE  Inv′
      BY ⟨2⟩1  DEF Next
    ⟨3⟩1.CASE IncreaseMaxBal(a, b)
      ⟨4⟩1. TypeOK′
        BY ⟨3⟩1  DEF TypeOK, IncreaseMaxBal
      ⟨4⟩2. VotesSafe′
        ⟨5⟩ SUFFICES ASSUME NEW a_1 ∈ Acceptor′, NEW b_1 ∈ Ballot′, NEW v ∈ Value′
                     PROVE  VotedFor(a_1, b_1, v)′ ⇒ SafeAt(b_1, v)′
          BY  DEF VotesSafe
        ⟨5⟩1. ∀ aa ∈ Acceptor, bb ∈ Ballot, vv ∈ Value :
                VotedFor(aa, bb, vv) ≡ VotedFor(aa, bb, vv)′
          BY ⟨3⟩1  DEF IncreaseMaxBal, VotedFor
        ⟨5⟩2. ∀ aa ∈ Acceptor, bb ∈ Ballot :
                maxBal[aa] > bb ⇒ maxBal′[aa] > bb
          BY ⟨3⟩1  DEF IncreaseMaxBal, TypeOK, Ballot
        ⟨5⟩3. ∀ aa ∈ Acceptor, bb ∈ Ballot :
                DidNotVoteAt(aa, bb) ⇒ DidNotVoteAt(aa, bb)′
          BY ⟨3⟩1  DEF IncreaseMaxBal, DidNotVoteAt, VotedFor
        ⟨5⟩4. ∀ aa ∈ Acceptor, bb ∈ Ballot :
                CannotVoteAt(aa, bb) ⇒ CannotVoteAt(aa, bb)′
          BY ⟨3⟩1, ⟨5⟩2, ⟨5⟩3  DEF IncreaseMaxBal, CannotVoteAt
        ⟨5⟩5. ∀ bb ∈ Ballot, vv ∈ Value :
                NoneOtherChoosableAt(bb, vv) ⇒ NoneOtherChoosableAt(bb, vv)′
          BY ⟨5⟩1, ⟨5⟩4, QuorumAssumptionDEFS NoneOtherChoosableAt
        ⟨5⟩6. QED
          BY ⟨5⟩1, ⟨5⟩5  DEF TypeOK, Ballot, VotesSafe, SafeAt
      ⟨4⟩3. OneValuePerBallot′
        BY ⟨3⟩1  DEF IncreaseMaxBal, OneValuePerBallot, VotedFor
      ⟨4⟩4. QED
        BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3  DEF Inv
    ⟨3⟩2. ASSUME NEW v ∈ Value,
                 VoteFor(a, b, v)
          PROVE  Inv′
      ⟨4⟩ SUFFICES ASSUME NEW Q ∈ Quorum,
                          ShowsSafeAt(Q, b, v)
                   PROVE  Inv′
        BY ⟨3⟩2  DEF VoteFor
      ⟨4⟩1. TypeOK′
        BY ⟨3⟩2  DEF TypeOK, VoteFor
      ⟨4⟩2. VotesSafe′  Using OneValuePerBallot in SafeAtStable
        ⟨5⟩ SUFFICES ASSUME NEW aa ∈ Acceptor′, NEW bb ∈ Ballot′, NEW vv ∈ Value′,
                            VotedFor(aa, bb, vv)′
                     PROVE  SafeAt(bb, vv)′
          BY  DEF VotesSafe
        ⟨5⟩1.CASE VotedFor(aa, bb, vv)
          ⟨6⟩1. SafeAt(bb, vv)
            BY ⟨5⟩1  DEF VotesSafe
          ⟨6⟩ QED
            BY ⟨4⟩1, ⟨6⟩1, SafeAtStable DEF Next
        ⟨5⟩2.CASE ¬VotedFor(aa, bb, vv)
          ⟨6⟩1. aa = a ∧ bb = b ∧ vv = v ∧ VotedFor(a, b, v)′
            BY ⟨3⟩2, ⟨4⟩1, ⟨5⟩2  DEF VoteFor, VotedFor, TypeOK
          ⟨6⟩ QED
            BY ⟨4⟩1, ⟨6⟩1, ShowsSafety, SafeAtStable DEF VoteFor, Next
        ⟨5⟩ QED
          BY ⟨5⟩1, ⟨5⟩2
      ⟨4⟩3. OneValuePerBallot′
        BY ⟨3⟩2  DEF VoteFor, OneValuePerBallot, VotedFor, TypeOK
      ⟨4⟩4. QED
        BY ⟨3⟩2, ⟨4⟩1, ⟨4⟩2, ⟨4⟩3  DEF Inv
    ⟨3⟩3. QED
      BY ⟨2⟩1, ⟨3⟩1, ⟨3⟩2
  ⟨2⟩2.CASE UNCHANGED ⟨votes, maxBal⟩
    BY ⟨2⟩2
    DEFS TypeOK, Next, VotesSafe, OneValuePerBallot,
          VotedFor, SafeAt, NoneOtherChoosableAt, CannotVoteAt, DidNotVoteAt,
          IncreaseMaxBal, VoteFor
  ⟨2⟩3. QED
    BY ⟨2⟩1, ⟨2⟩2
⟨1⟩3. QED
  BY ⟨1⟩1, ⟨1⟩2, PTL DEF Spec
├──────────────────────────────────────────────────────────────────────┤
C ≜ INSTANCE Consensus

THEOREM Spec ∧ Inv ⇒ C!Spec
⟨1⟩1. Init ⇒ C!Init
  BY QuorumAssumption, SetExtensionality, IsaM("force")
   DEF Init, C!Init, chosen, ChosenAt, VotedFor
⟨1⟩2. Next ∧ Inv ⇒ C!Next ∨ UNCHANGED chosen
  ⟨2⟩1 SUFFICES ASSUME Next, InvPROVE  C!Next ∨ UNCHANGED chosen
    BY ⟨2⟩1
  ⟨2⟩2. chosen ⊆ chosen′
    BY ⟨2⟩1, QuorumAssumption, Z3    SMTT(10) fails
     DEF Next, Inv, TypeOK, IncreaseMaxBal, chosen, ChosenAt, VotedFor, Ballot, VoteFor
  ⟨2⟩3. chosen′ = {} ∨ ∃ v ∈ Value : chosen′ = {v}
    ⟨3⟩1. PICK a ∈ Acceptor, b ∈ Ballot :
             ∨ IncreaseMaxBal(a, b)
             ∨ ∃ v ∈ Value : VoteFor(a, b, v)
      BY ⟨2⟩1  DEF Next
    ⟨3⟩2.CASE IncreaseMaxBal(a, b)
    ⟨3⟩3.CASE ∃ v ∈ Value : VoteFor(a, b, v)
    ⟨3⟩q. QED
      BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, SMT
  ⟨2⟩q. QED
    BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3, OneVoteThm, VotesSafeImpliesConsistency, SetExtensionality, SMT
     DEF Inv, C!Next
⟨1⟩3. QED
  PROOF OMITTED
```