```
1  ---------- MODULE Voting ----------
2  EXTENDS Sets
3  ----------------------------------
4  CONSTANT Value, Acceptor, Quorum

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

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

13 Ballot ≜ Nat
14 ----------------------------------
15 VARIABLES votes, maxBal

17 TypeOK ≜ ∧ votes ∈ [Acceptor → SUBSET (Ballot × Value)]
18          ∧ maxBal ∈ [Acceptor → Ballot ∪ {−1}]
19 ----------------------------------
20 VotedFor(a, b, v) ≜ ⟨b, v⟩ ∈ votes[a]

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

24 ShowsSafeAt(Q, b, v) ≜
25   ∧ ∀ a ∈ Q : maxBal[a] ≥ b  have promised
26   ∧ ∃ c ∈ −1 .. (b − 1) :
27       ∧ (c ≠ −1) ⇒ ∃ a ∈ Q : VotedFor(a, c, v)
28       ∧ ∀ d ∈ (c + 1) .. (b − 1), a ∈ Q : DidNotVoteAt(a, d)
29 ----------------------------------
30 Init ≜
31     ∧ votes = [a ∈ Acceptor ↦ {}]
32     ∧ maxBal = [a ∈ Acceptor ↦ −1]

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

39 VoteFor(a, b, v) ≜
40     ∧  maxBal[a] ≤ b  keep promise
41     ∧  ∀ vt ∈ votes[a] : vt[1] ≠ b
42     ∧  ∀ c ∈ Acceptor \ {a} :
43          ∀ vt ∈ votes[c] : (vt[1] = b) ⇒ (vt[2] = v)
44     ∧  ∃ Q ∈ Quorum : ShowsSafeAt(Q, b, v)  safe to vote
45     ∧  votes′ = [votes EXCEPT ![a] = votes[a] ∪ {⟨b, v⟩}]  vote
46     ∧  maxBal′ = [maxBal EXCEPT ![a] = b]  make promise
47 ----------------------------------
```

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

53  Spec ≜ Init ∧ □[Next]⟨votes, maxBal⟩
54  ----------------------------------------------------------------
55  ChosenAt(b, v) ≜
56      ∃ Q ∈ Quorum : ∀ a ∈ Q : VotedFor(a, b, v)

58  chosen ≜ {v ∈ Value : ∃ b ∈ Ballot : ChosenAt(b, v)}

60  Consistency ≜ chosen = {} ∨ ∃ v ∈ Value : chosen = {v}   Cardinality(chosen) ≤ 1
61  ----------------------------------------------------------------
62  CannotVoteAt(a, b) ≜
63      ∧ maxBal[a] > b
64      ∧ DidNotVoteAt(a, b)

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

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

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

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

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

85  Inv ≜ TypeOK ∧ VotesSafe ∧ OneValuePerBallot
86  ----------------------------------------------------------------
87  THEOREM AllSafeAtZero ≜ ∀ v ∈ Value : SafeAt(0, v)
88    BY DEF SafeAt

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

95  THEOREM OneVoteThm ≜ OneValuePerBallot ⇒ OneVote
```

```
96    BY DEF OneValuePerBallot, OneVote
97  ⊢------------------------------------------------------------⊣
98  THEOREM VotesSafeImpliesConsistency ≜
99     ASSUME VotesSafe, OneVote, chosen ≠ {}
100    PROVE  ∃ v ∈ Value : chosen = {v}
101 ⟨1⟩1. PICK v ∈ Value : v ∈ chosen
102   BY DEF chosen
103 ⟨1⟩2. SUFFICES ASSUME NEW w ∈ chosen
104                  PROVE  w = v
105   BY ⟨1⟩1, ⟨1⟩2
106 ⟨1⟩3. ASSUME NEW b1 ∈ Ballot, NEW b2 ∈ Ballot, b1 < b2,
107                NEW v1 ∈ Value, NEW v2 ∈ Value,
108                ChosenAt(b1, v1) ∧ ChosenAt(b2, v2)
109       PROVE  v1 = v2
110   ⟨2⟩1. SafeAt(b2, v2)
111     BY ⟨1⟩3, QuorumAssumption, SMT DEF ChosenAt, VotesSafe
112   ⟨2⟩2. QED
113     BY ⟨1⟩3, ⟨2⟩1, QuorumAssumption, Z3
114     DEFS CannotVoteAt, DidNotVoteAt, OneVote,
115          ChosenAt, NoneOtherChoosableAt, Ballot, SafeAt
116 ⟨1⟩4. QED
117   BY QuorumAssumption, ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, Z3
118   DEFS Ballot, ChosenAt, OneVote, chosen

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

128 THEOREM SafeAtStable ≜ Inv ∧ Next ∧ TypeOK′ ⇒
129                                ∀ b ∈ Ballot, v ∈ Value :
130                                  SafeAt(b, v) ⇒ SafeAt(b, v)′
131   OMITTED
132 ⊢------------------------------------------------------------⊣
133 THEOREM Invariant ≜ Spec ⇒ □Inv
134 ⟨1⟩ USE DEF Inv
135 ⟨1⟩1. Init ⇒ Inv
136   BY DEF Init, TypeOK, VotesSafe, OneValuePerBallot, VotedFor
137 ⟨1⟩2. Inv ∧ [Next]⟨votes, maxBal⟩ ⇒ Inv′
138   ⟨2⟩ SUFFICES ASSUME Inv, [Next]⟨votes, maxBal⟩
139                PROVE  Inv′
140     OBVIOUS
```

⟨2⟩1.CASE *Next*
⟨3⟩ SUFFICES ASSUME NEW $a \in Acceptor$, NEW $b \in Ballot$,
$\lor\ IncreaseMaxBal(a,\ b)$
$\lor \exists\, v \in Value : VoteFor(a,\ b,\ v)$
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 $a\_1 \in Acceptor'$, NEW $b\_1 \in Ballot'$, NEW $v \in Value'$
PROVE $VotedFor(a\_1,\ b\_1,\ v)' \Rightarrow SafeAt(b\_1,\ v)'$
BY DEF *VotesSafe*
⟨5⟩1. $\forall\, aa \in Acceptor,\ bb \in Ballot,\ vv \in Value :$
$VotedFor(aa,\ bb,\ vv) \equiv VotedFor(aa,\ bb,\ vv)'$
BY ⟨3⟩1 DEF *IncreaseMaxBal*, *VotedFor*
⟨5⟩2. $\forall\, aa \in Acceptor,\ bb \in Ballot :$
$maxBal[aa] > bb \Rightarrow maxBal'[aa] > bb$
BY ⟨3⟩1 DEF *IncreaseMaxBal*, *TypeOK*, *Ballot*
⟨5⟩3. $\forall\, aa \in Acceptor,\ bb \in Ballot :$
$DidNotVoteAt(aa,\ bb) \Rightarrow DidNotVoteAt(aa,\ bb)'$
BY ⟨3⟩1 DEF *IncreaseMaxBal*, *DidNotVoteAt*, *VotedFor*
⟨5⟩4. $\forall\, aa \in Acceptor,\ bb \in Ballot :$
$CannotVoteAt(aa,\ bb) \Rightarrow CannotVoteAt(aa,\ bb)'$
BY ⟨3⟩1, ⟨5⟩2, ⟨5⟩3 DEF *IncreaseMaxBal*, *CannotVoteAt*
⟨5⟩5. $\forall\, bb \in Ballot,\ vv \in Value :$
$NoneOtherChoosableAt(bb,\ vv) \Rightarrow NoneOtherChoosableAt(bb,\ vv)'$
BY ⟨5⟩1, ⟨5⟩4, *QuorumAssumption*DEFS *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 $v \in Value$,
$VoteFor(a,\ b,\ v)$
PROVE Inv'
⟨4⟩ SUFFICES ASSUME NEW $Q \in Quorum$,
$ShowsSafeAt(Q,\ b,\ v)$
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 $aa \in Acceptor'$, NEW $bb \in Ballot'$, NEW $vv \in Value'$,

```
                                    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
```

```
THEOREM Consistent ≜ Spec ⇒ □Consistency
  OMITTED
```

```
C ≜ INSTANCE Consensus WITH chosen ← chosen

THEOREM Refinement ≜ Spec ⇒ C!Spec
⟨1⟩1. Init ⇒ C!Init
  BY QuorumAssumption, SetExtensionality, IsaM("force")
   DEF Init, C!Init, chosen, ChosenAt, VotedFor
⟨1⟩2. TypeOK' ∧ Consistency' ∧ [Next]⟨votes, maxBal⟩ ⇒ [C!Next]chosen
  ⟨2⟩1. UNCHANGED ⟨votes, maxBal⟩ ⇒ UNCHANGED chosen
    BY  DEF chosen, ChosenAt, VotedFor
  ⟨2⟩2. TypeOK' ∧ Consistency' ∧ Next ⇒ C!Next ∨ UNCHANGED chosen
    ⟨3⟩1. SUFFICES ASSUME TypeOK', Consistency', Next
```

PROVE $C!Next \lor$ UNCHANGED $chosen$

OBVIOUS

$\langle 3 \rangle 2$. $chosen \subseteq chosen'$

BY $\langle 3 \rangle 1$, $QuorumAssumption$, $Z3$

DEFS $Next$, $IncreaseMaxBal$, $VoteFor$, Inv, $TypeOK$, $chosen$, $ChosenAt$, $VotedFor$, $Ballot$

$\langle 3 \rangle 3$. $chosen' = \{\} \lor \exists\, v \in Value : chosen' = \{v\}$

BY $\langle 3 \rangle 1$ DEF $Consistency$

$\langle 3 \rangle 4$. QED

BY $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, $\langle 3 \rangle 3$ DEF $C!Next$

$\langle 2 \rangle 3$. QED

BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$

$\langle 1 \rangle 3$. QED

BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $Invariant$, $Consistent$, PTL DEF $Spec$, Inv, $C!Spec$