```
1  ---------- MODULE Voting ----------
   This is a high-level algorithm in which a set of processes cooperatively choose a value.
6  EXTENDS Integers
7  ----------
8  CONSTANTS
9      Value,      The set of choosable values.
10     Acceptor,   A set of processes that will choose a value.
11     Quorum      The set of "quorums", where a "quorum" is a "large enough" set of acceptors.
13 ASSUME QuorumAssumption ≜
14     ∧  ∀ Q ∈ Quorum : Q ⊆ Acceptor
15     ∧  ∀ Q1, Q2 ∈ Quorum : Q1 ∩ Q2 ≠ {}
17 THEOREM QuorumNonEmpty ≜ ∀ Q ∈ Quorum : Q ≠ {}
18 ----------
19 Ballot ≜ Nat  The set of "ballot numbers".
20 ----------
   Each acceptor can cast one or more votes, where each vote cast by an acceptor has the form ⟨b, v⟩
   indicating that the acceptor has voted for value v in ballot b.
26 VARIABLES
27     votes,    votes[a]: the set of votes cast by acceptor a
28     maxBal    maxBal[a]: a ballot number.
29               Acceptor a will cast further votes only in ballots numbered ≥ maxBal[a].
31 TypeOK ≜
32     ∧ votes ∈ [Acceptor → SUBSET (Ballot × Value)]
33     ∧ maxBal ∈ [Acceptor → Ballot ∪ {−1}]
34 ----------
35 VotedFor(a, b, v) ≜ ⟨b, v⟩ ∈ votes[a]  Acceptor a has voted for v in ballot b.
37 ChosenAt(b, v) ≜  ⟨b, v⟩ is chosen if a quorum of acceptors have voted for it.
38     ∃ Q ∈ Quorum :
39         ∀ a ∈ Q : VotedFor(a, b, v)
41 chosen ≜  The set of values that have been chosen.
42     {v ∈ Value : ∃ b ∈ Ballot : ChosenAt(b, v)}
43 ----------
44 DidNotVoteAt(a, b) ≜  The acceptor a did not vote (for any value) at ballot b.
45     ∀ v ∈ Value : ¬ VotedFor(a, b, v)
47 CannotVoteAt(a, b) ≜  The acceptor a cannot vote (for any value) at ballot b.
48     ∧ DidNotVoteAt(a, b)
49     ∧ maxBal[a] > b
51 NoneOtherChoosableAt(b, v) ≜  ChosenAt(b, w) is not and can never become true for any w ≠ v.
52     ∃ Q ∈ Quorum :
53         ∀ a ∈ Q : VotedFor(a, b, v) ∨ CannotVoteAt(a, b)
```

```
55 THEOREM ChoosableThm ≜
56              ∀ b ∈ Ballot, v ∈ Value :
57                ChosenAt(b, v) ⇒ NoneOtherChoosableAt(b, v)
58 ----------------------------------------------------------------------------
59 SafeAt(b, v) ≜  No value other than v has been or can ever be chosen in any ballot < b.
60     ∀ c ∈ 0 .. (b − 1) : NoneOtherChoosableAt(c, v)

62 THEOREM AllSafeAtZero ≜ ∀ v ∈ Value : SafeAt(0, v)
63 ----------------------------------------------------------------------------
64 OneVote ≜ ∀ a ∈ Acceptor, b ∈ Ballot, v, w ∈ Value :
65               VotedFor(a, b, v) ∧ VotedFor(a, b, w) ⇒ (v = w)

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

71 THEOREM OneValuePerBallot ⇒ OneVote
72 ----------------------------------------------------------------------------
73 VotesSafe ≜ ∀ a ∈ Acceptor, b ∈ Ballot, v ∈ Value :
74                 VotedFor(a, b, v) ⇒ SafeAt(b, v)

76 THEOREM VotesSafeImpliesConsistency ≜
77              ∧ TypeOK
78              ∧ VotesSafe
79              ∧ OneVote
80              ⇒ ∨ chosen = {}
81                ∨ ∃ v ∈ Value : chosen = {v}
82 ----------------------------------------------------------------------------
83 ShowsSafeAt(Q, b, v) ≜
84     ∧ ∀ a ∈ Q : maxBal[a] ≥ b
85     ∧ ∃ c ∈ −1 .. (b − 1) :
86         ∧ (c ≠ −1) ⇒ ∃ a ∈ Q : VotedFor(a, c, v)
87         ∧ ∀ d ∈ (c + 1) .. (b − 1), a ∈ Q : DidNotVoteAt(a, d)

89 THEOREM ShowsSafety ≜
90              TypeOK ∧ VotesSafe ∧ OneValuePerBallot ⇒
91                ∀ Q ∈ Quorum, b ∈ Ballot, v ∈ Value :
92                  ShowsSafeAt(Q, b, v) ⇒ SafeAt(b, v)
93 ----------------------------------------------------------------------------
94 Init ≜
95     ∧ votes = [a ∈ Acceptor ↦ {}]
96     ∧ maxBal = [a ∈ Acceptor ↦ −1]

98 IncreaseMaxBal(a, b) ≜  Acceptor a is allowed ot increase maxBal[a] to a ballot number.
99     ∧ b > maxBal[a]
100    ∧ maxBal′ = [maxBal EXCEPT ![a] = b]
101    ∧ UNCHANGED votes
```

$VoteFor(a,\ b,\ v) \triangleq$ Acceptor a votes for v in ballot b.
  $\land\ maxBal[a] \leq b$ The acceptor cannot cast a vote in a ballot less than $maxBal[a]$
  $\land\ \forall\, vt \in votes[a] : vt[1] \neq b$ to maintain *OneValuePerBallot*
  $\land\ \forall\, c \in Acceptor \setminus \{a\} :$
    $\forall\, vt \in votes[c] : (vt[1] = b) \Rightarrow (vt[2] = v)$
  $\land\ \exists\, Q \in Quorum : ShowsSafeAt(Q,\ b,\ v)$ to maintain *VotesSafe*
  $\land\ votes' = [votes \text{ EXCEPT } ![a] = @ \cup \{\langle b,\ v\rangle\}]$
  $\land\ maxBal' = [maxBal \text{ EXCEPT } ![a] = b]$

$Next \triangleq$
  $\exists\, a \in Acceptor,\ b \in Ballot :$
    $\lor\ IncreaseMaxBal(a,\ b)$
    $\lor \exists\, v \in Value : VoteFor(a,\ b,\ v)$

$Spec \triangleq Init \land \Box[Next]_{\langle votes,\ maxBal\rangle}$

$Inv \triangleq TypeOK \land VotesSafe \land OneValuePerBallot$

THEOREM $Invariance \triangleq Spec \Rightarrow \Box Inv$

$C \triangleq$ INSTANCE $Consensus$ WITH $Value \leftarrow Value,\ chosen \leftarrow chosen$

THEOREM $Spec \Rightarrow C!Spec$
$\langle 1\rangle 1.\ Inv \land Init \Rightarrow C!Init$
$\langle 1\rangle 2.\ Inv \land [Next]_{\langle votes,\ maxBal\rangle} \Rightarrow [C!Next]_{chosen}$
$\langle 1\rangle 3.$ QED
  $\langle 2\rangle 1. \Box Inv \land \Box[Next]_{\langle votes,\ maxBal\rangle} \Rightarrow \Box[C!Next]_{chosen}$
    BY $\langle 1\rangle 2$ and temporal reasoning
  $\langle 2\rangle 2. \Box Inv \land Spec \Rightarrow C!Spec$
    BY $\langle 2\rangle 1,\ \langle 1\rangle 1$
  $\langle 2\rangle 3.$ QED
    BY $\langle 2\rangle 2,\ Invariance$