```
MODULE UniversalPaxosStoreWithVotes
Extend UniversalPaxosStore with an explicit record of votes that have been accepted by participants. This is used to demonstrate that UniversalPaxosStore refines EagerVoting.
EXTENDS UniversalPaxosStore, TLAPS

VARIABLE votes

TypeOKV ≜
    ∧ TypeOK
    ∧ votes ∈ [Participant → SUBSET (Ballot × Value)]

InitV ≜
    ∧ Init
    ∧ votes = [p ∈ Participant ↦ {}]

PrepareV(p, b) ≜
    ∧ Prepare(p, b)
    ∧ UNCHANGED votes

UpdateStateV(q, p, pp) ≜
    ∧ UpdateState(q, p, pp)
    ∧ IF state[q][q].maxBal ≤ pp.maxVBal ∧ pp.maxVBal ≠ −1 accept
      THEN votes' = [votes EXCEPT ![q] = @ ∪ {⟨pp.maxVBal, pp.maxVVal⟩}]
      ELSE UNCHANGED votes

OnMessageV(q) ≜
  ∃ m ∈ msgs :
    ∧ q ∈ m.to
    ∧ LET p ≜ m.from
        IN  UpdateStateV(q, p, m.state[p]) replacing UpdateState
    ∧ IF ∨ m.state[q].maxBal < state'[q][q].maxBal
         ∨ m.state[q].maxVBal < state'[q][q].maxVBal
      THEN Send([from ↦ q, to ↦ {m.from}, state ↦ state'[q]])
      ELSE UNCHANGED msgs

AcceptV(p, b, v) ≜
    ∧  Accept(p, b, v)
    ∧  votes' = [votes EXCEPT ![p] = @ ∪ {⟨b, v⟩}] accept

NextV ≜ ∃ p ∈ Participant : ∨ OnMessageV(p)
                            ∨ ∃ b ∈ Ballot : ∨ PrepareV(p, b)
                                             ∨ ∃ v ∈ Value : AcceptV(p, b, v)
SpecV ≜ InitV ∧ □[NextV]⟨vars, votes⟩

THEOREM Invariant ≜ SpecV ⇒ □TypeOKV
  OMITTED
```

```
    UniversalPaxosStore refines EagerVoting.
54  maxBal ≜ [p ∈ Participant ↦ state[p][p].maxBal]

56  EV ≜ INSTANCE EagerVoting WITH Acceptor ← Participant

58  THEOREM SpecV ⇒ EV!Spec
59    ⟨1⟩1. InitV       ⇒ EV!Init
60      BY DEF InitV, Init, EV!Init, InitState, maxBal
61    ⟨1⟩2. TypeOKV′ ∧ [NextV]⟨vars, votes⟩ ⇒ [EV!Next]⟨votes, maxBal⟩
62      ⟨2⟩1. UNCHANGED ⟨state, msgs, votes⟩ ⇒ UNCHANGED ⟨votes, maxBal⟩
63        BY DEF maxBal
64      ⟨2⟩2. TypeOKV′ ∧ NextV ⇒ EV!Next ∨ UNCHANGED ⟨votes, maxBal⟩
65        ⟨3⟩.SUFFICES ASSUME TypeOKV′, NextV
66                     PROVE  EV!Next ∨ UNCHANGED ⟨votes, maxBal⟩
67          OBVIOUS
68        ⟨3⟩1. ASSUME NEW p ∈ Participant,
69                     OnMessageV(p)
70              PROVE  EV!Next ∨ UNCHANGED ⟨votes, maxBal⟩
71        ⟨3⟩2. ASSUME NEW p ∈ Participant,
72                     NEW b ∈ Ballot,
73                     PrepareV(p, b)
74              PROVE  EV!Next
75          ⟨4⟩1. EV!IncreaseMaxBal(p, b)
76            BY ⟨3⟩2 DEF EV!IncreaseMaxBal, PrepareV, Prepare, Ballot, maxBal, TypeOKV, TypeOK, State
77          ⟨4⟩2. QED
78            BY ⟨3⟩2, ⟨4⟩1 DEF EV!Next, EV!Ballot, Ballot
      BY ⟨3⟩2 DEF TypeOKV, EV!TypeOK, TypeOK, EV!Next, EV!IncreaseMaxBal,
      EV!Ballot, PrepareV, Prepare, Ballot, maxBal
82        ⟨3⟩3. ASSUME NEW p ∈ Participant,
83                     NEW b ∈ Ballot,
84                     NEW v ∈ Value,
85                     AcceptV(p, b, v)
86              PROVE  EV!Next ∨ UNCHANGED ⟨votes, maxBal⟩
87        ⟨3⟩4. QED
88          BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3 DEF NextV
89      ⟨2⟩3. QED
90        BY ⟨2⟩1, ⟨2⟩2 DEF vars
91    ⟨1⟩3. QED
92      BY ⟨1⟩1, ⟨1⟩2, Invariant, PTL DEF SpecV, EV!Spec
93  ====================================================================

    \ * Modification History
    \ * Last modified Wed Aug 14 22:09:16 CST 2019 by hengxin
    \ * Created Wed Aug 14 14:05:06 CST 2019 by hengxin
```