```
──────────────────── MODULE CASPaxos ────────────────────
This is a high-level specification of the CASPaxos algorithm from the paper "CASPaxos: Repli-
cated State Machines without Logs" by Denis Rystsov.

Please go to https://arxiv.org/abs/1802.07000 for the paper.

This spec is adapted from that of Paxos consensus algorithm by Leslie Lamport, which can be
found at https://github.com/tlaplus/Examples/blob/master/specifications/PaxosHowToWinATuringAward/Paxos.tla.

Search "⟨ + ⟩" for the code added for CASPaxos.

TODO: It refines the spec in module Voting.

EXTENDS Integers
─────────────────────────────────────────────────────────
CONSTANTS
    Value,        the set of values to be proposed and chosen from
    Acceptor,     the set acceptors
    Quorum        the quorum system on acceptors

None ≜ CHOOSE v : v ∉ Value

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

⟨ + ⟩ The set of all possible CAS operations. The CAS operations with cmpVal = None are
initialization operations. We assume that the new values (i.e., swapVal) are not None.
CASOperation ≜ [cmpVal : Value ∪ {None}, swapVal : Value]

Message ≜  the set of all possible messages that can be sent in the algorithm
        [type : {"1a"}, bal : Ballot]
    ∪   [type : {"1b"}, acc : Acceptor, bal : Ballot,
         mbal : Ballot ∪ { − 1}, mval : Value ∪ {None}]
    ∪   [type : {"2a"}, bal : Ballot, val : Value]
    ∪   [type : {"2b"}, acc : Acceptor, bal : Ballot, val : Value]
    ∪   [type : {"response"}, bal : Ballot]   ⟨ + ⟩ the messages sent to the user
─────────────────────────────────────────────────────────
VARIABLES
     maxBal[a]: the last ballot the acceptor a ∈ Acceptor has voted for
    maxBal,
     ⟨maxVBal[a], maxVVal[a]⟩ is the vote with the largest ballot cast by acceptor a ∈ Acceptor.
     It equals ⟨ − 1, None⟩ if a ∈ Acceptor has not cast any vote.
    maxVBal, maxVVal,
    msgs,       the set of all messages that have been sent
    ops         ⟨ + ⟩ops[b]: the CAS operation to be proposed at ballot b ∈ Ballot

vars ≜ ⟨maxBal, maxVBal, maxVVal, msgs, ops⟩
```

```
────────────────────────────────────────────────────────────
TypeOK ≜ ∧ maxBal   ∈ [Acceptor → Ballot ∪ {−1}]
         ∧ maxVBal  ∈ [Acceptor → Ballot ∪ {−1}]
         ∧ maxVVal  ∈ [Acceptor → Value ∪ {None}]
         ∧ msgs ⊆ Message
         ∧ ops ∈ [Ballot → CASOperation]  ⟨+⟩
────────────────────────────────────────────────────────────
Init ≜ ∧ maxBal  = [a ∈ Acceptor ↦ −1]
       ∧ maxVBal = [a ∈ Acceptor ↦ −1]
       ∧ maxVVal = [a ∈ Acceptor ↦ None]
       ∧ msgs = {}
        ⟨+⟩ops remains unchanged; we utilize TLC to explore all possible CAS operations.
       ∧ ops ∈ [Ballot → CASOperation]
────────────────────────────────────────────────────────────
Send(m) ≜ msgs′ = msgs ∪ {m}
────────────────────────────────────────────────────────────
```

The leader of ballot $b \in Ballot$ sends a $Phase1a$ message.

```
Phase1a(b) ≜
   ∧  Send([type ↦ "1a", bal ↦ b])
   ∧  UNCHANGED ⟨maxBal, maxVBal, maxVVal, ops⟩
```

The acceptor $a \in Acceptor$ receives a $Phase1a$ message and sends back a $Phase1b$ message.

For refinement: This action implements the $IncreaseMaxBal(a, b)$ action of the Voting algorithm for $b = m.bal$.

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

In the $Phase2a(b, v)$ action, the ballot b leader sends a type "2a" message asking the acceptors to vote for some value computed based on v in ballot number b.

For refinement: the enabling conditions of the action–its first two conjuncts–ensure that the second through fourth conjuncts of the four enabling conditions of action $VoteFor(a, b, v)$ in module Voting will be true when acceptor a receives that message.

```
Phase2a(b, v) ≜
  ∧ ¬∃ m ∈ msgs  : m.type = "2a" ∧ m.bal = b
  ∧ ∃ Q ∈ Quorum :
       LET Q1b ≜ {m ∈ msgs  : ∧ m.type = "1b"
                               ∧ m.acc ∈ Q
                               ∧ m.bal = b}
           Q1bv ≜ {m ∈ Q1b : m.mbal ≥ 0}
       IN   ∧ ∀ a ∈ Q : ∃ m ∈ Q1b : m.acc = a
```

```
                    $\land \lor \land Q1bv = \{\}$   $\langle + \rangle CAS(None, v)$ as an initialization operation
                            $\land ops[b].cmpVal = None$   $\langle + \rangle$
                        $\lor \exists m \in Q1bv :$   $\langle + \rangle CAS(v, ops[b].swapVal)$ as an atomic compare-and-swap operation
                                $\land m.mval = v$
                                $\land \forall mm \in Q1bv : m.mbal \geq mm.mbal$
                                $\land ops[b].cmpVal = v$   $\langle + \rangle$
    $\land Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto ops[b].swapVal])$   $\langle + \rangle val \mapsto ops[b].swapVal$
    $\land \text{UNCHANGED } \langle maxBal, maxVBal, maxVVal, ops \rangle$
```

The $Phase2b(a)$ action describes what $a \in Acceptor$ does when it receives a phase $2a$ message $m \in msgs$, which is sent by the leader of ballot $m.bal$ asking acceptors to vote for $m.val$ in that ballot.

For refinement: The enabling condition of the $Phase2b(a)$ action together with the receipt of the phase $2a$ message m implies that the $VoteFor(a, m.bal, m.val)$ action of module Voting is enabled and can be executed.

```
$Phase2b(a) \triangleq$
    $\land \exists m \in msgs :$
        $\land m.type = \text{"2a"}$
        $\land m.bal \geq maxBal[a]$
        $\land maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$
        $\land maxVBal' = [maxVBal \text{ EXCEPT } ![a] = m.bal]$
        $\land maxVVal' = [maxVVal \text{ EXCEPT } ![a] = m.val]$
        $\land Send([type \mapsto \text{"2b"}, acc \mapsto a, bal \mapsto m.bal, val \mapsto m.val])$
    $\land \text{UNCHANGED } \langle ops \rangle$
```

$\langle + \rangle$ The leader of ballot $b \in Ballot$ responds to the user.

```
$Respond(b) \triangleq$
    $\land \neg \exists m \in msgs : m.type = \text{"response"} \land m.bal = b$
    $\land \exists Q \in Quorum :$
        $\text{LET } Q2b \triangleq \{m \in msgs : \land m.type = \text{"2b"}$
                                        $\land m.acc \in Q$
                                        $\land m.bal = b\}$
        $\text{IN } \forall a \in Q : \exists m \in Q2b : m.acc = a$
    $\land Send([type \mapsto \text{"response"}, bal \mapsto b])$
    $\land \text{UNCHANGED } \langle maxBal, maxVBal, maxVVal, ops \rangle$
```

```
$Next \triangleq \lor \exists b \in Ballot : \lor Phase1a(b)$
                                  $\lor \exists v \in Value : Phase2a(b, v)$
                                  $\lor Respond(b)$   $\langle + \rangle$
              $\lor \exists a \in Acceptor : \lor Phase1b(a)$
                                      $\lor Phase2b(a)$

$Spec \triangleq Init \land \Box[Next]_{vars}$
```
