MODULE *CASPaxos*

This is a high-level specification of the *CASPaxos* algorithm from the paper "*CASPaxos*: Replicated State Machines without Logs" by *Denis Rystsov*.

The *CASPaxos* algorithm implements a linearizable *CAS* register. Note that since linearizability is a local property, it is sufficient to model a single *CAS* register in a system.

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 "$\langle + \rangle$" 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

$None \triangleq \text{CHOOSE } v : v \notin Value$

ASSUME $\land \forall\, Q \in Quorum : Q \subseteq Acceptor$
$\land \forall\, Q1,\, Q2 \in Quorum : Q1 \cap Q2 \neq \{\}$

$Ballot \triangleq Nat$

$\langle + \rangle$ 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*.

$CASOperation \triangleq [cmpVal : Value \cup \{None\},\, swapVal : Value]$

$Message \triangleq$ the set of all possible messages that can be sent in the algorithm
- $[type : \{\text{"1a"}\},\, bal : Ballot]$
- $\cup\ [type : \{\text{"1b"}\},\, acc : Acceptor,\, bal : Ballot,\, mbal : Ballot \cup \{-1\},\, mval : Value \cup \{None\}]$
- $\cup\ [type : \{\text{"2a"}\},\, bal : Ballot,\, val : Value]$
- $\cup\ [type : \{\text{"2b"}\},\, acc : Acceptor,\, bal : Ballot,\, val : Value]$
- $\cup\ [type : \{\text{"response"}\},\, bal : Ballot]$ $\langle + \rangle$ the messages sent to the user

VARIABLES

$maxBal$[a]: the last ballot the acceptor $a \in Acceptor$ has voted for

maxBal,

$\langle maxVBal[a],\, maxVVal[a] \rangle$ is the vote with the largest ballot cast by acceptor $a \in Acceptor$. It equals $\langle -1,\, None \rangle$ if $a \in Acceptor$ has not cast any vote.

maxVBal, *maxVVal*,

msgs, the set of all messages that have been sent

ops $\langle + \rangle ops[b]$: the *CAS* operation to be proposed at ballot $b \in Ballot$

$vars \triangleq \langle maxBal,\ maxVBal,\ maxVVal,\ msgs,\ ops \rangle$

$TypeOK \triangleq \land\ maxBal \in [Acceptor \to Ballot \cup \{-1\}]$
$\land\ maxVBal \in [Acceptor \to Ballot \cup \{-1\}]$
$\land\ maxVVal \in [Acceptor \to Value \cup \{None\}]$
$\land\ msgs \subseteq Message$
$\land\ ops \in [Ballot \to CASOperation]$ $\langle + \rangle$

$Init \triangleq \land\ maxBal = [a \in Acceptor \mapsto -1]$
$\land\ maxVBal = [a \in Acceptor \mapsto -1]$
$\land\ maxVVal = [a \in Acceptor \mapsto None]$
$\land\ msgs = \{\}$
$\langle + \rangle ops$ remains unchanged; we utilize *TLC* to explore all possible *CAS* operations.
$\land\ ops \in [Ballot \to CASOperation]$

$Send(m) \triangleq msgs' = msgs \cup \{m\}$

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

$Phase1a(b) \triangleq$
$\land$ $Send([type \mapsto$ "1a", $bal \mapsto b])$
$\land$ UNCHANGED $\langle maxBal,\ maxVBal,\ maxVVal,\ ops \rangle$

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 $b = m.bal$.

$Phase1b(a) \triangleq$
$\land \exists\, m \in msgs :$
$\land\ m.type =$ "1a"
$\land\ m.bal > maxBal[a]$
$\land\ maxBal' = [maxBal$ EXCEPT $![a] = m.bal]$
$\land\ Send([type \mapsto$ "1b", $acc \mapsto a,\ bal \mapsto m.bal,$
$mbal \mapsto maxVBal[a],\ mval \mapsto maxVVal[a]])$
$\land$ UNCHANGED $\langle maxVBal,\ maxVVal,\ ops \rangle$

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.

$Phase2a(b,\ v) \triangleq$
$\land \neg\exists\, m \in msgs \ : m.type =$ "2a" $\land\ m.bal = b$
$\land \exists\, Q \in Quorum :$
LET $Q1b \triangleq \{m \in msgs \ : \land\ m.type =$ "1b"
$\land\ m.acc \in Q$
$\land\ m.bal = b\}$
$Q1bv \triangleq \{m \in Q1b : m.mbal \geq 0\}$

```
111         IN      ∧ ∀ a ∈ Q : ∃ m ∈ Q1b : m.acc = a
112                 ∧ ∨ ∧ Q1bv = {}   ⟨+⟩CAS(None, v) as an initialization operation
113                     ∧ ops[b].cmpVal = None   ⟨+⟩
114                   ∨ ∃ m ∈ Q1bv :   ⟨+⟩CAS(v, ops[b].swapVal) as an atomic compare-and-swap operation
115                       ∧ m.mval = v
116                       ∧ ∀ mm ∈ Q1bv : m.mbal ≥ mm.mbal
117                       ∧ ops[b].cmpVal = v   ⟨+⟩ not all CAS operations will terminate due to this precondition
118     ∧ Send([type ↦ "2a", bal ↦ b, val ↦ ops[b].swapVal])   ⟨+⟩val ↦ ops[b].swapVal
119     ∧ UNCHANGED ⟨maxBal, maxVBal, maxVVal, ops⟩
```

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.

```
131 Phase2b(a) ≜
132     ∧ ∃ m ∈ msgs :
133         ∧ m.type = "2a"
134         ∧ m.bal ≥ maxBal[a]
135         ∧ maxBal' = [maxBal EXCEPT ![a] = m.bal]
136         ∧ maxVBal' = [maxVBal EXCEPT ![a] = m.bal]
137         ∧ maxVVal' = [maxVVal EXCEPT ![a] = m.val]
138         ∧ Send([type ↦ "2b", acc ↦ a, bal ↦ m.bal, val ↦ m.val])
139     ∧ UNCHANGED ⟨ops⟩
```

⟨+⟩ The leader of ballot $b \in Ballot$ responds to the user.

```
143 Respond(b) ≜
144     ∧ ¬∃ m ∈ msgs   : m.type = "response" ∧ m.bal = b
145     ∧ ∃ Q ∈ Quorum :
146         LET Q2b ≜ {m ∈ msgs : ∧ m.type = "2b"
147                               ∧ m.acc ∈ Q
148                               ∧ m.bal  = b}
149         IN   ∀ a ∈ Q : ∃ m ∈ Q2b : m.acc = a
150     ∧ Send([type ↦ "response", bal ↦ b])
151     ∧ UNCHANGED ⟨maxBal, maxVBal, maxVVal, ops⟩
152 ├──────────────────────────────────────────────────────────────────────┤
153 Next ≜
154     ∨ ∃ b ∈ Ballot :
155         ∨ Phase1a(b)
156         ∨ ∃ v ∈ Value : Phase2a(b, v)
157         ∨ Respond(b)   ⟨+⟩
158     ∨ ∃ a ∈ Acceptor :
159         ∨ Phase1b(a)
160         ∨ Phase2b(a)

162 Spec ≜ Init ∧ □[Next]_vars
```

```
163 └──────────────────────────────────────────────────────────────────────┘
    \ * Modification History
    \ * Last modified Wed Jul 27 09:47:34 CST 2022 by hengxin
    \ * Created Tue Jul 20 23:30:00 CST 2022 by hengxin
```