MODULE *TPaxos*

EXTENDS *Integers*, *FiniteSets*

CONSTANTS

$\quad Participant,$ the set of partipants

$\quad Value,$ the set of possible input values for *Participant* to propose

$\quad Quorum$

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

$NP \triangleq Cardinality(Participant)$ number of $p \in Participant$

ASSUME $QuorumAssumption \triangleq$

$\quad \land \ \forall Q \in Quorum : Q \subseteq Participant$

$\quad \land \ \forall Q1, Q2 \in Quorum : Q1 \cap Q2 \neq \{\}$

$Ballot \triangleq Nat$

$Max(m, n) \triangleq \text{IF } m > n \text{ THEN } m \text{ ELSE } n$

$Injective(f) \triangleq \forall a, b \in \text{DOMAIN } f : (a \neq b) \Rightarrow (f[a] \neq f[b])$

$PIndex \triangleq \text{CHOOSE } f \in [Participant \rightarrow 1 \ldots NP] : Injective(f)$

$Bals(p) \triangleq \{b \in Ballot : b\%NP = PIndex[p] - 1\}$ allocate ballots for each $p \in Participant$

$State \triangleq [maxBal : Ballot \cup \{-1\},$

$\qquad\qquad maxVBal : Ballot \cup \{-1\}, maxVVal : Value \cup \{None\}]$

$InitState \triangleq [maxBal \mapsto -1, maxVBal \mapsto -1, maxVVal \mapsto None]$

$Message \triangleq [from : Participant, to : \text{SUBSET } Participant, state : [Participant \rightarrow State]]$

VARIABLES

$\quad state,$ $state[p][q]$: the state of $q \in Participant$ from the view of $p \in Participant$

$\quad msgs$ the set of messages that have been sent

$vars \triangleq \langle state, msgs \rangle$

$TypeOK \triangleq$

$\quad \land \ state \in [Participant \rightarrow [Participant \rightarrow State]]$

$\quad \land \ msgs \subseteq Message$

$Init \triangleq$

$\quad \land state = [p \in Participant \mapsto [q \in Participant \mapsto InitState]]$

$\quad \land msgs = \{\}$

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

$p \in Participant$ starts the prepare phase by issuing a ballot $b \in Ballot$.

$Prepare(p, b) \triangleq$

$\quad \land \ b \in Bals(p)$

$\quad \land \ state[p][p].maxBal < b$

$\quad \land \ state' = [state \text{ EXCEPT } ![p][p].maxBal = b]$

```
    $\land\ Send([from \mapsto p,\ to \mapsto Participant,\ state \mapsto state'[p]])$
```

$q \in Participant$ updates its own state $state[q]$ according to the actual state pp of $p \in Participant$ extracted from a message $m \in Message$ it receives. This is called by $OnMessage(q)$ of $TPaxosAP$.

Note: pp is $m.state[p]$; it may not be equal to $state[p][p]$ at the time $UpdateState$ is called.

```
$UpdateState(q, p, pp) \triangleq$
    $state' = [state$ EXCEPT
            $![q][p].maxBal = Max(@, pp.maxBal),$
            $![q][p].maxVBal = Max(@, pp.maxVBal),$
            $![q][p].maxVVal =$ IF $state[q][p].maxVBal < pp.maxVBal$
                                THEN $pp.maxVVal$ ELSE $@,$
            $![q][q].maxBal = Max(@, pp.maxBal),$    make promise
            $![q][q].maxVBal =$ IF $state[q][q].maxBal \leq pp.maxVBal$  accept
                                THEN $pp.maxVBal$ ELSE $@,$
            $![q][q].maxVVal =$ IF $state[q][q].maxBal \leq pp.maxVBal$  accept
                                THEN $pp.maxVVal$ ELSE $@]$
```

$q \in Participant$ receives and processes a message in $Message$.

```
$OnMessage(q) \triangleq$
    $\exists\, m \in msgs:$
        $\land\ q \in m.to$
        $\land$ LET $p \triangleq m.from$
          IN  $UpdateState(q, p, m.state[p])$
        $\land$ LET $qm \triangleq [from \mapsto m.from,\ to \mapsto m.to \setminus \{q\},\ state \mapsto m.state]$  remove $q$ from to
              $nm \triangleq [from \mapsto q,\ to \mapsto \{m.from\},\ state \mapsto state'[q]]$  new message to reply
          IN  IF $\lor\ m.state[q].maxBal < state'[q][q].maxBal$
                 $\lor\ m.state[q].maxVBal < state'[q][q].maxVBal$
              THEN $msgs' = (msgs \setminus \{m\}) \cup \{qm, nm\}$
              ELSE $msgs' = (msgs \setminus \{m\}) \cup \{qm\}$
```

```
$p \in Participant$ starts the accept phase by issuing the ballot $b \in Ballot$ with value $v \in Value$.
$Accept(p, b, v) \triangleq$
    $\land b \in Bals(p)$
    $\land state[p][p].maxBal \leq b$    corresponding to the first conjunction in $Voting$
    $\land state[p][p].maxVBal \neq b$   correspongding to the second conjunction in $Voting$
    pick $v$ based on a quorum of $state[p]$
    $\land \exists Q \in Quorum :$   collecting "enough" replies to $Prepare(p, b)$
      $\land \forall q \in Q : state[p][q].maxBal = b$
      $\land \lor \forall q \in Q : state[p][q].maxVBal = -1$   free to pick its own value
        $\lor \exists q \in Q :$   $v$ is the value with the highest $maxVBal$
             $\land state[p][q].maxVVal = v$
             $\land \forall r \in Participant : state[p][q].maxVBal \geq state[p][r].maxVBal$
    pick $v$ based on $state[p]$
    $\land \lor \forall q \in Participant : state[p][q].maxVBal = -1$ \ * free to pick its own value
    $\lor \exists q \in Participant :$ \ * $v$ is the value with the highest $maxVBal$
       $\land state[p][q].maxVVal = v$
       $\land \forall r \in Participant: state[p][q].maxVBal \geq state[p][r].maxVBal$
    $\land state' = [state$ EXCEPT $![p][p].maxVBal = b,$
                                  $![p][p].maxVVal = v]$
    $\land Send([from \mapsto p, to \mapsto Participant, state \mapsto state'[p]])$

$Next \triangleq \exists p \in Participant : \lor OnMessage(p)$
                                      $\lor \exists b \in Ballot : \lor Prepare(p, b)$
                                                          $\lor \exists v \in Value : Accept(p, b, v)$
$Spec \triangleq Init \land \Box[Next]_{vars}$

$ChosenP(p) \triangleq$   the set of values chosen by $p \in Participant$
    $\{v \in Value : \exists b \in Ballot :$
                 $\exists Q \in Quorum : \forall q \in Q : \land state[p][q].maxVBal = b$
                                               $\land state[p][q].maxVVal = v\}$

$chosen \triangleq$ UNION $\{ChosenP(p) : p \in Participant\}$

$Consistency \triangleq Cardinality(chosen) \leq 1$

THEOREM $Spec \Rightarrow \Box Consistency$
```

\ * Modification History
\ * Last modified Sun *Sep* 08 13:06:23 *CST* 2019 by *pure_*
\ * Last modified *Wed Jul* 31 15:00:12 *CST* 2019 by *hengxin*
\ * Last modified *Mon Jun* 03 21:26:09 *CST* 2019 by *stary*
\ * Last modified *Wed* May 09 21:39:31 *CST* 2018 by dell
\ * Created *Mon Apr* 23 15:47:52 *GMT* +08:00 2018 by *pure_*