MODULE *BasicPaxos*

EXTENDS *Integers*, *FiniteSets*

$Maximum(S) \triangleq \text{IF } S = \{\} \text{ THEN } -1$
$\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE CHOOSE } n \in S : \forall\, m \in S : n \geq m$

CONSTANTS *Acceptors*, *Ballot*, *Value*

$NoneValue \triangleq \text{CHOOSE } v : v \notin Value$

$Messages \triangleq [type : \{\text{“prepare”}\},\ bal : Ballot]$
$\qquad\qquad\qquad\qquad \cup$
$\qquad\qquad\qquad\qquad [type : \{\text{“promise”}\},\ bal : Ballot,\ maxVBal : Ballot \cup \{-1\},\ maxVVal : Value \cup \{NoneValue\},\ ac$
$\qquad\qquad\qquad\qquad \cup$
$\qquad\qquad\qquad\qquad [type : \{\text{“accept”}\},\ \ bal : Ballot,\ val : Value]$
$\qquad\qquad\qquad\qquad \cup$
$\qquad\qquad\qquad\qquad [type : \{\text{“accepted”}\},\ maxVBal : Ballot,\ maxVVal : Value,\ acc : Acceptors]$

$Quorums \triangleq \{Q \in \text{SUBSET } Acceptors : Cardinality(Q) * 2 > Cardinality(Acceptors)\}$

ASSUME $\land\, Ballot \subseteq Nat$
$\qquad\land\, 0 \in Ballot$
$\qquad\land\, \forall\, Q1,\ Q2 \in Quorums : Q1 \cap Q2 \neq \{\}$

VARIABLES *state*, *msgs*

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

$TypeOK \triangleq \land\, state \in [Acceptors \to [maxBal : Ballot \cup \{-1\},$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad maxVBal : Ballot \cup \{-1\},$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad maxVVal : Value \cup \{NoneValue\}]]$
$\qquad\qquad\land\, msgs \subseteq Messages$

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

$Init \triangleq \land\, state = [a \in Acceptors \mapsto [maxBal \mapsto -1,\ maxVBal \mapsto -1,\ maxVVal \mapsto NoneValue]]$
$\qquad\land\, msgs = \{\}$

$Prepare(b) \triangleq \land \neg\exists\, m \in msgs : m.type = \text{“prepare”} \land m.bal = b$
$\qquad\qquad\land\, Send([type \mapsto \text{“prepare”},\ bal \mapsto b])$
$\qquad\qquad\land\, \text{UNCHANGED } state$

$Promise(acc) \triangleq \exists\, msg \in msgs : \land\, msg.type = \text{“prepare”}$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\land\, state[acc].maxBal < msg.bal$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\land\, state' = [state \text{ EXCEPT } ![acc].maxBal = msg.bal]$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\land\, Send([type \quad \mapsto \text{“promise”},$

$$\begin{array}{ll} bal & \mapsto msg.bal, \\ maxVBal & \mapsto state[acc].maxVBal, \\ maxVVal & \mapsto state[acc].maxVVal, \\ acc & \mapsto acc]) \end{array}$$

$$\begin{array}{l} Accept(b) \triangleq \ \land \neg \exists\, m \in msgs : m.type = \text{“accept”} \land m.bal = b \\ \qquad \land \exists\, Q \in Quorums : \\ \qquad\quad \text{LET } mset \triangleq \{m \in msgs : \ \land m.type = \text{“promise”} \\ \qquad\qquad\qquad\qquad\qquad\quad \land m.bal = b \\ \qquad\qquad\qquad\qquad\qquad\quad \land m.acc \in Q\} \\ \qquad\qquad mu \triangleq Maximum(\{m.maxVBal : m \in mset\}) \\ \qquad\qquad v \triangleq \text{IF } mu = -1 \text{ THEN CHOOSE } val \in Value : \text{TRUE} \\ \qquad\qquad\qquad\qquad\qquad\quad \text{ELSE } (\text{CHOOSE } m \in mset : m.maxVBal = mu).maxVVal \\ \qquad\quad \text{IN} \quad \land \forall\, ac \in Q : \exists\, m \in mset : m.acc = ac \\ \qquad\qquad\quad \land Send([type \mapsto \text{“accept”},\ bal \mapsto b,\ val \mapsto v]) \\ \qquad \land \text{UNCHANGED } state \end{array}$$

$$\begin{array}{l} Accepted(acc) \triangleq \exists\, msg \in msgs : \ \land msg.type = \text{“accept”} \\ \qquad\qquad\qquad\qquad\qquad\quad \land state[acc].maxBal \leq msg.bal \\ \qquad\qquad\qquad\qquad\qquad\quad \land state[acc].maxVBal < msg.bal \\ \qquad\qquad\qquad\qquad\qquad\quad \land state' = [state \text{ EXCEPT } ![acc].maxBal = msg.bal, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![acc].maxVBal = msg.bal, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![acc].maxVVal = msg.val] \\ \qquad\qquad\qquad\qquad\qquad\quad \land Send([type \mapsto \text{“accepted”},\ maxVBal \mapsto msg.bal,\ maxVVal \mapsto msg.val,\ ac \end{array}$$

$$\begin{array}{l} Next \triangleq \ \lor \exists\, b \in Ballot : Prepare(b) \lor Accept(b) \\ \qquad\quad \lor \exists\, a \in Acceptors : Promise(a) \lor Accepted(a) \end{array}$$

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

$$\begin{array}{l} VoteForIn(a,\ v,\ b) \triangleq \exists\, m \in msgs : \ \land m.type = \text{“accepted”} \\ \qquad\qquad\qquad\qquad\qquad\quad \land m.maxVVal = v \\ \qquad\qquad\qquad\qquad\qquad\quad \land m.maxVBal = b \\ \qquad\qquad\qquad\qquad\qquad\quad \land m.acc = a \end{array}$$

There exists a quorum accepting the $proposal(b,\ v)$

$$\begin{array}{l} ChosenIn(v,\ b) \triangleq \exists\, Q \in Quorums : \\ \qquad\qquad\qquad\quad \forall\, a \in Q : VoteForIn(a,\ v,\ b) \end{array}$$

$$Chosen(v) \triangleq \exists\, b \in Ballot : ChosenIn(v,\ b)$$

Only a value is chosen in a ballot

$$Consistency \triangleq \forall\, v1,\ v2 \in Value : Chosen(v1) \land Chosen(v2) \Rightarrow (v1 = v2)$$

$$\begin{array}{l} ChosenSet \triangleq \{v \in Value : \exists\, b \in Ballot : \\ \qquad\qquad\qquad\qquad\quad \exists\, Q \in Quorums : \forall\, a \in Q : \ \land state[a].maxVBal = b \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \land state[a].maxVVal = v\} \end{array}$$

There exists some value being chosen eventually
And it should be false, because *Paxos* does not satisfy liveness

$$Liveness \triangleq \Diamond(ChosenSet \neq \{\})$$

\ * Modification History
\ * Last modified *Wed Jan* 13 20:14:22 *CST* 2021 by Dell
\ * Created *Wed Jan* 13 17:10:09 *CST* 2021 by Dell