MODULE *HPaxos*

EXTENDS *Naturals*, *FiniteSets*, *Functions*

CONSTANT *LastBallot*
ASSUME $LastBallot \in Nat$

$Ballot \triangleq Nat$

CONSTANT *Value*
ASSUME $ValueNotEmpty \triangleq Value \neq \{\}$

CONSTANTS *SafeAcceptor*,
FakeAcceptor,
ByzQuorum,
Learner

$Acceptor \triangleq SafeAcceptor \cup FakeAcceptor$

ASSUME $AcceptorAssumption \triangleq$
$\land \quad SafeAcceptor \cap FakeAcceptor = \{\}$
$\land\ Acceptor \cap Learner = \{\}$

ASSUME $BQAssumption \triangleq$
$\land \quad SafeAcceptor \in ByzQuorum$
$\land \quad \forall\, Q \in ByzQuorum : Q \subseteq Acceptor$

Learner graph

CONSTANT *TrustLive*
ASSUME $TrustLiveAssumption \triangleq$
$TrustLive \in \text{SUBSET } [lr : Learner,\ q : ByzQuorum]$

CONSTANT *TrustSafe*
ASSUME $TrustSafeAssumption \triangleq$
$TrustSafe \in \text{SUBSET } [from : Learner,\ to : Learner,\ q : ByzQuorum]$

ASSUME $LearnerGraphAssumptionSymmetry \triangleq$
$\forall\, E \in TrustSafe :$
$[from \mapsto E.to,\ to \mapsto E.from,\ q \mapsto E.q] \in TrustSafe$

ASSUME $LearnerGraphAssumptionTransitivity \triangleq$
$\forall\, E1,\ E2 \in TrustSafe :$
$E1.q = E2.q \land E1.to = E2.from \Rightarrow$
$[from \mapsto E1.from,\ to \mapsto E2.to,\ q \mapsto E1.q] \in TrustSafe$

ASSUME $LearnerGraphAssumptionClosure \triangleq$

$\forall\, E \in \mathit{TrustSafe} : \forall\, Q \in \mathit{ByzQuorum} :$
$E.q \subseteq Q \Rightarrow$
$[\mathit{from} \mapsto E.\mathit{from},\ \mathit{to} \mapsto E.\mathit{to},\ q \mapsto Q] \in \mathit{TrustSafe}$

ASSUME $\mathit{LearnerGraphAssumptionValidity} \triangleq$
$\forall\, E \in \mathit{TrustSafe} : \forall\, Q1,\ Q2 \in \mathit{ByzQuorum} :$
$[\mathit{lr} \mapsto E.\mathit{from},\ q \mapsto Q1] \in \mathit{TrustLive} \land$
$[\mathit{lr} \mapsto E.\mathit{to},\ q \mapsto Q2] \in \mathit{TrustLive} \Rightarrow$
$\exists\, N \in E.q : N \in Q1 \land N \in Q2$

Entanglement relation

$\mathit{Ent} \triangleq \{LL \in \mathit{Learner} \times \mathit{Learner} :$
$[\mathit{from} \mapsto LL[1],\ \mathit{to} \mapsto LL[2],\ q \mapsto \mathit{SafeAcceptor}] \in \mathit{TrustSafe}\}$

Messages

CONSTANT $\mathit{MaxRefCardinality}$
ASSUME $\mathit{MaxRefCardinalityAssumption} \triangleq$
$\land \quad \mathit{MaxRefCardinality} \in \mathit{Nat}$
$\land \quad \mathit{MaxRefCardinality} \geq 1$

$\mathit{RefCardinality} \triangleq \mathit{Nat}$
$\mathit{RefCardinality} \triangleq 1\,..\,\mathit{MaxRefCardinality}$

$\mathit{FINSUBSET}(S,\ R) \triangleq \{\mathit{Range}(\mathit{seq}) : \mathit{seq} \in [R \to S]\}$
$\mathit{FINSUBSET}(S,\ K) \triangleq \{\mathit{Range}(\mathit{seq}) : \mathit{seq} \in [1\,..\,K \to S]\}$
$\mathit{FINSUBSET}(S,\ R) \triangleq \text{UNION } \{\{\mathit{Range}(\mathit{seq}) : \mathit{seq} \in [1\,..\,K \to S]\} : K \in R\}$

$\mathit{MessageRec}0 \triangleq$
$[\mathit{type} : \{\text{“1a”}\},\ \mathit{bal} : \mathit{Ballot},\ \mathit{ref} : \{\{\}\}]$

$\mathit{MessageRec}1(M,\ n) \triangleq$
$M \cup$
$[\mathit{type} : \{\text{“1b”}\},$
$\mathit{acc} : \mathit{Acceptor},$
$\mathit{ref} : \mathit{FINSUBSET}(M,\ \mathit{RefCardinality})] \cup$
$[\mathit{type} : \{\text{“2a”}\},$
$\mathit{lrn} : \mathit{Learner},$
$\mathit{acc} : \mathit{Acceptor},$
$\mathit{ref} : \mathit{FINSUBSET}(M,\ \mathit{RefCardinality})]$

$\mathit{MessageRec}[n \in \mathit{Nat}] \triangleq$
IF $n = 0$
THEN $\mathit{MessageRec}0$
ELSE $\mathit{MessageRec}1(\mathit{MessageRec}[n-1],\ n)$

CONSTANT $\mathit{MaxMessageDepth}$

ASSUME $MaxMessageDepth \in Nat$

$MessageDepthRange \triangleq Nat$

$Message \triangleq \text{UNION } \{MessageRec[n] : n \in MessageDepthRange\}$

Non-message value

$None \triangleq \text{CHOOSE } v : v \notin Message$
$NoMessage \triangleq None$

Transitive references

Bounded transitive references

$TranBound0 \triangleq [m \in Message \mapsto \{m\}]$
$TranBound1(tr,\ n) \triangleq$
  $[m \in Message \mapsto \{m\} \cup \text{UNION } \{tr[r] : r \in m.ref\}]$

$TranBound[n \in Nat] \triangleq$
  IF $n = 0$
  THEN $TranBound0$
  ELSE $TranBound1(TranBound[n-1],\ n)$

Countable transitive references

$TranDepthRange \triangleq MessageDepthRange$

$Tran(m) \triangleq \text{UNION } \{TranBound[n][m] : n \in TranDepthRange\}$

Algorithm specification

TODO comment

VARIABLES $msgs$,
  $known\_msgs$,
  $recent\_msgs$,
  $queued\_msg$,
  $2a\_lrn\_loop$,
  $processed\_lrns$,
  $decision$,
  $BVal$

$Get1a(m) \triangleq$
  $\{x \in Tran(m) :$
    $\land\ x.type = \text{“1a”}$
    $\land\ \forall\, y \in Tran(m) :$
      $y.type = \text{“1a”} \Rightarrow y.bal \leq x.bal\}$

$B(m,\ bal) \triangleq \exists\, x \in Get1a(m) : bal = x.bal$

$V(m,\ val) \triangleq \exists\, x \in Get1a(m) : val = BVal[x.bal]$

Maximal ballot number of any messages known to acceptor a

$MaxBal(a,\ mbal) \triangleq$
$\quad \land \exists\, m \in known\_msgs[a] : B(m,\ mbal)$
$\quad \land \forall\, x \in known\_msgs[a] :$
$\qquad \forall\, b \in Ballot : B(x,\ b) \Rightarrow b \leq mbal$

$SameBallot(x,\ y) \triangleq$
$\quad \forall\, b \in Ballot : B(x,\ b) \equiv B(y,\ b)$

The acceptor is _*caught*_ in a message x if the transitive references of x include evidence such as two messages both signed by the acceptor, in which neither is featured in the other's transitive references.

$CaughtMsg(x) \triangleq$
$\quad \{m \in Tran(x) :$
$\qquad \land\, m.type \neq$ "1a"
$\qquad \land \exists\, m1 \in Tran(x) :$
$\qquad\quad \land\, m1.type \neq$ "1a"
$\qquad\quad \land\, m.acc = m1.acc$
$\qquad\quad \land\, m \notin Tran(m1)$
$\qquad\quad \land\, m1 \notin Tran(m)\}$

$Caught(x) \triangleq \{m.acc : m \in CaughtMsg(x)\}$

Connected

$ConByQuorum(a,\ b,\ x,\ S) \triangleq$ a : *Learner*, *b* : *Learner*, x : 1*b*, $S \in ByzQuorum$
$\quad \land\, [from \mapsto a,\ to \mapsto b,\ q \mapsto S] \in TrustSafe$
$\quad \land\, S \cap Caught(x) = \{\}$

$Con(a,\ x) \triangleq$ a : *Learner*, x : 1*b*
$\quad \{b \in Learner :$
$\qquad \exists\, S \in ByzQuorum : ConByQuorum(a,\ b,\ x,\ S)\}$

2*a*-message is _*buried*_ if there exists a quorum of acceptors that have seen 2*a*-messages with different values, the same learner, and higher ballot numbers.

$Buried(x,\ y) \triangleq$ x : 2*a*, y : 1*b*
$\quad$ LET $Q \triangleq \{m \in Tran(y) :$
$\qquad\qquad \exists\, z \in Tran(m) :$
$\qquad\qquad\quad \land\, z.type =$ "2a"
$\qquad\qquad\quad \land\, z.lrn = x.lrn$
$\qquad\qquad\quad \land \forall\, bx,\ bz \in Ballot :$
$\qquad\qquad\qquad B(x,\ bx) \land B(z,\ bz) \Rightarrow bx < bz$
$\qquad\qquad\quad \land \forall\, vx,\ vz \in Value :$

$$V(x,\, vx) \land V(z,\, vz) \Rightarrow vx \neq vz\}$$
IN $\quad [lr \mapsto x.lrn,\; q \mapsto \{m.acc : m \in Q\}] \in TrustLive$

Connected $2a$ messages

$Con2as(l,\, x) \triangleq$ $\quad l : Learner,\, x : 1b$

$\{m \in Tran(x) :$
$\land\, m.type = \text{“2a”}$
$\land\, m.acc = x.acc$
$\land\, \neg Buried(m,\, x)$
$\land\, m.lrn \in Con(l,\, x)\}$

Fresh $1b$ messages

$Fresh(l,\, x) \triangleq$ $\quad l : Learner,\, x : 1b$

$\forall\, m \in Con2as(l,\, x) : \forall\, v \in Value : V(x,\, v) \equiv V(m,\, v)$

Quorum of messages referenced by $2a$

$q(x) \triangleq$ $\quad x : 2a$

LET $Q \triangleq \{m \in Tran(x) :$
$\land\, m.type = \text{“1b”}$
$\land\, Fresh(x.lrn,\, m)$
$\land\, \forall\, b \in Ballot : B(m,\, b) \equiv B(x,\, b)\}$
IN $\quad \{m.acc : m \in Q\}$

$WellFormed(m) \triangleq$
$\land\, m \in Message$
$\land\, \exists\, b \in Ballot : B(m,\, b)$ *TODO* prove it
$\land\, m.type = \text{“1b”} \Rightarrow$
$\forall\, y \in Tran(m) :$
$m \neq y \land SameBallot(m,\, y) \Rightarrow y.type = \text{“1a”}$
$\land\, m.type = \text{“2a”} \Rightarrow$
$\land\, [lr \mapsto m.lrn,\; q \mapsto q(m)] \in TrustLive$

$vars \triangleq \langle msgs,\, known\_msgs,\, recent\_msgs,\, queued\_msg,$
$2a\_lrn\_loop,\, processed\_lrns,\, decision,\, BVal\rangle$

$Init \triangleq$
$\land\, msgs = \{\}$
$\land\, known\_msgs = [x \in Acceptor \cup Learner \mapsto \{\}]$
$\land\, recent\_msgs = [a \in Acceptor \cup Learner \mapsto \{\}]$
$\land\, queued\_msg = [a \in Acceptor \mapsto NoMessage]$
$\land\, 2a\_lrn\_loop = [a \in Acceptor \mapsto \text{FALSE}]$
$\land\, processed\_lrns = [a \in Acceptor \mapsto \{\}]$
$\land\, decision = [lb \in Learner \times Ballot \mapsto \{\}]$
$\land\, BVal \in [Ballot \to Value]$

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

$Proper(a,\ m) \triangleq \forall\, r \in m.ref : r \in known\_msgs[a]$

$Recv(a,\ m) \triangleq$
$\quad \land\ m \notin known\_msgs[a]$ ignore known messages
$\quad \land\ WellFormed(m)$
$\quad \land\ Proper(a,\ m)$

$Store(a,\ m) \triangleq$
$\quad \land\ known\_msgs' = [known\_msgs \text{ EXCEPT } ![a] = known\_msgs[a] \cup \{m\}]$

$Send1a(b) \triangleq$
$\quad \land\ Send([type \mapsto \text{“1a”},\ bal \mapsto b,\ ref \mapsto \{\}])$
$\quad \land$ UNCHANGED $\langle known\_msgs,\ recent\_msgs,\ queued\_msg,$
$\qquad\qquad 2a\_lrn\_loop,\ processed\_lrns,\ decision\rangle$
$\quad \land$ UNCHANGED $BVal$

$Known2a(l,\ b,\ v) \triangleq$
$\quad \{x \in known\_msgs[l] :$
$\qquad \land\ x.type = \text{“2a”}$
$\qquad \land\ x.lrn = l$
$\qquad \land\ B(x,\ b)$
$\qquad \land\ V(x,\ v)\}$

The following is invariant for *queued_msg* variable values.
For any safe acceptor A, if $queued\_msg[A] \neq NoMessage$ then
$queued\_msg[A]$ is a well-formed message of type “1b” sent by A,
having the direct references all known to A.

$Process1a(a,\ m) \triangleq$
$\quad$ LET $new1b \triangleq [type \mapsto \text{“1b”},\ acc \mapsto a,\ ref \mapsto recent\_msgs[a] \cup \{m\}]$ IN
$\quad \land\ m.type = \text{“1a”}$
$\quad \land\ Recv(a,\ m)$
$\quad \land\ Store(a,\ m)$
$\quad \land\ WellFormed(new1b) \Rightarrow$
$\qquad \land\ Send(new1b)$
$\qquad \land\ recent\_msgs' = [recent\_msgs \text{ EXCEPT } ![a] = \{\}]$
$\qquad \land\ queued\_msg' = [queued\_msg \text{ EXCEPT } ![a] = new1b]$
$\quad \land\ (\neg WellFormed(new1b)) \Rightarrow$
$\qquad \land\ recent\_msgs' = [recent\_msgs \text{ EXCEPT } ![a] = recent\_msgs[a] \cup \{m\}]$
$\qquad \land$ UNCHANGED $\langle msgs,\ queued\_msg\rangle$
$\quad \land$ UNCHANGED $\langle 2a\_lrn\_loop,\ processed\_lrns,\ decision\rangle$
$\quad \land$ UNCHANGED $BVal$

$Process1b(a,\ m) \triangleq$
$\quad \land\ m.type = \text{“1b”}$
$\quad \land\ Recv(a,\ m)$
$\quad \land\ Store(a,\ m)$

$\land\ recent\_msgs' = [recent\_msgs \text{ EXCEPT } ![a] = recent\_msgs[a] \cup \{m\}]$
$\land\ (\forall\, mb,\ b \in Ballot : MaxBal(a,\ mb) \land B(m,\ b) \Rightarrow mb \leq b) \Rightarrow$
$\quad \land\ 2a\_lrn\_loop' = [2a\_lrn\_loop \text{ EXCEPT } ![a] = \text{TRUE}]$
$\quad \land\ processed\_lrns' = [processed\_lrns \text{ EXCEPT } ![a] = \{\}]$
$\land\ (\neg(\forall\, mb,\ b \in Ballot : MaxBal(a,\ mb) \land B(m,\ b) \Rightarrow mb \leq b)) \Rightarrow$
$\quad \text{UNCHANGED } \langle 2a\_lrn\_loop,\ processed\_lrns\rangle$
$\land\ queued\_msg' = [queued\_msg \text{ EXCEPT } ![a] = NoMessage]$
$\land\ \text{UNCHANGED } \langle msgs,\ decision\rangle$
$\land\ \text{UNCHANGED } BVal$

$Process1bLearnerLoopStep(a,\ lrn) \triangleq$
$\text{LET } new2a \triangleq [type \mapsto \text{“2a”},\ lrn \mapsto lrn,\ acc \mapsto a,\ ref \mapsto recent\_msgs[a]] \text{IN}$
$\land\ processed\_lrns' =$
$\quad [processed\_lrns \text{ EXCEPT } ![a] = processed\_lrns[a] \cup \{lrn\}]$
$\land\ WellFormed(new2a) \Rightarrow$
$\quad \land\ Send(new2a)$
$\quad \land\ Store(a,\ new2a)$
$\quad \land\ recent\_msgs' = [recent\_msgs \text{ EXCEPT } ![a] = \{new2a\}]$
$\land\ (\neg WellFormed(new2a)) \Rightarrow$
$\quad \text{UNCHANGED } \langle msgs,\ known\_msgs,\ recent\_msgs\rangle$
$\land\ \text{UNCHANGED } \langle queued\_msg,\ 2a\_lrn\_loop,\ decision\rangle$
$\land\ \text{UNCHANGED } BVal$

$Process1bLearnerLoopDone(a) \triangleq$
$\land\ Learner \setminus processed\_lrns[a] = \{\}$
$\land\ 2a\_lrn\_loop' = [2a\_lrn\_loop \text{ EXCEPT } ![a] = \text{FALSE}]$
$\land\ \text{UNCHANGED } \langle msgs,\ known\_msgs,\ recent\_msgs,\ queued\_msg,$
$\qquad\qquad processed\_lrns,\ decision\rangle$
$\land\ \text{UNCHANGED } BVal$

$Process1bLearnerLoop(a) \triangleq$
$\lor\ \exists\, lrn \in Learner \setminus processed\_lrns[a] :$
$\quad Process1bLearnerLoopStep(a,\ lrn)$
$\lor\ Process1bLearnerLoopDone(a)$

$Process2a(a,\ m) \triangleq$
$\land\ m.type = \text{“2a”}$
$\land\ Recv(a,\ m)$
$\land\ Store(a,\ m)$
$\land\ recent\_msgs' = [recent\_msgs \text{ EXCEPT } ![a] = recent\_msgs[a] \cup \{m\}]$
$\land\ \text{UNCHANGED } \langle msgs,\ queued\_msg,\ 2a\_lrn\_loop,\ processed\_lrns,\ decision\rangle$
$\land\ \text{UNCHANGED } BVal$

$ProposerSendAction \triangleq$
$\exists\, bal \in Ballot : Send1a(bal)$

$AcceptorProcessAction \triangleq$

```
    ∃ a ∈ SafeAcceptor :
      ∨ ∧ 2a_lrn_loop[a] = FALSE
        ∧ ∨ ∧ queued_msg[a] ≠ NoMessage
              ∧ Process1b(a, queued_msg[a])
          ∨ ∧ queued_msg[a] = NoMessage
              ∧ ∃ m ∈ msgs :
                  ∧ ∨ Process1a(a, m)
                    ∨ Process1b(a, m)
      ∨ ∧ 2a_lrn_loop[a] = TRUE
        ∧ Process1bLearnerLoop(a)

FakeSend1b(a) ≜
    ∧ ∃ fin ∈ FINSUBSET(msgs, RefCardinality) :
        LET new1b ≜ [type ↦ "1b", acc ↦ a, ref ↦ fin] IN
          ∧ WellFormed(new1b)
          ∧ Send(new1b)
    ∧ UNCHANGED ⟨known_msgs, recent_msgs, queued_msg,
                 2a_lrn_loop, processed_lrns, decision⟩
    ∧ UNCHANGED BVal

FakeSend2a(a) ≜
    ∧ ∃ fin ∈ FINSUBSET(msgs, RefCardinality) :
        ∃ lrn ∈ Learner :
          LET new2a ≜ [type ↦ "2a", lrn ↦ lrn, acc ↦ a, ref ↦ fin] IN
            ∧ WellFormed(new2a)
            ∧ Send(new2a)
    ∧ UNCHANGED ⟨known_msgs, recent_msgs, queued_msg,
                 2a_lrn_loop, processed_lrns, decision⟩
    ∧ UNCHANGED BVal

LearnerRecv(l, m) ≜
    ∧ Recv(l, m)
    ∧ Store(l, m)
    ∧ UNCHANGED ⟨msgs, recent_msgs, queued_msg,
                 2a_lrn_loop, processed_lrns, decision⟩
    ∧ UNCHANGED BVal

ChosenIn(l, b, v) ≜
    ∃ S ∈ SUBSET Known2a(l, b, v) :
      [lr ↦ l, q ↦ {m.acc : m ∈ S}] ∈ TrustLive

LearnerDecide(l, b, v) ≜
    ∧ ChosenIn(l, b, v)
    ∧ decision' = [decision EXCEPT ![⟨l, b⟩] = decision[l, b] ∪ {v}]
    ∧ UNCHANGED ⟨msgs, known_msgs, recent_msgs, queued_msg,
                 2a_lrn_loop, processed_lrns⟩
```

$\land$ UNCHANGED $BVal$

$LearnerAction \triangleq$
$\exists\, lrn \in Learner :$
$\lor \exists\, m \in msgs :$
$LearnerRecv(lrn,\, m)$
$\lor \exists\, bal \in Ballot :$
$\exists\, val \in Value :$
$LearnerDecide(lrn,\, bal,\, val)$

$FakeAcceptorAction \triangleq$
$\exists\, a \in FakeAcceptor :$
$\lor FakeSend1b(a)$
$\lor FakeSend2a(a)$

$Next \triangleq$
$\lor ProposerSendAction$
$\lor AcceptorProcessAction$
$\lor LearnerAction$
$\lor FakeAcceptorAction$

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

$Safety \triangleq$
$\forall\, L1,\, L2 \in Learner : \forall\, B1,\, B2 \in Ballot : \forall\, V1,\, V2 \in Value :$
$\langle L1,\, L2\rangle \in Ent \land$
$V1 \in decision[L1,\, B1] \land V2 \in decision[L2,\, B2] \Rightarrow$
$V1 = V2$

THEOREM $SafetyResult \triangleq Spec \Rightarrow \Box Safety$

Sanity check propositions

$SanityCheck0 \triangleq$
$\forall\, L \in Learner : Cardinality(known\_msgs[L]) = 0$

$SanityCheck1 \triangleq$
$\forall\, L \in Learner : \forall\, m1,\, m2 \in known\_msgs[L] :$
$\forall\, b1,\, b2 \in Ballot :$
$B(m1,\, b1) \land B(m2,\, b2) \Rightarrow b1 = b2$

$2aNotSent \triangleq$
$\forall\, M \in msgs : M.type \neq$ "2a"

$2aNotSentBySafeAcceptor \triangleq$
$\forall\, M \in msgs : M.type =$ "2a" $\Rightarrow M.acc \notin SafeAcceptor$

$1b\mathit{NotSentBySafeAcceptor} \triangleq$
$\forall\, M \in \mathit{msgs} : M.\mathit{type} = \text{“1b”} \Rightarrow M.\mathit{acc} \notin \mathit{SafeAcceptor}$

$\mathit{NoDecision} \triangleq$
$\forall\, L \in \mathit{Learner} : \forall\, \mathit{BB} \in \mathit{Ballot} : \forall\, \mathit{VV} \in \mathit{Value} :$
$\mathit{VV} \notin \mathit{decision}[L,\, \mathit{BB}]$

$\mathit{UniqueDecision} \triangleq$
$\forall\, L1,\, L2 \in \mathit{Learner} : \forall\, B1,\, B2 \in \mathit{Ballot} : \forall\, V1,\, V2 \in \mathit{Value} :$
$V1 \in \mathit{decision}[L1,\, B1] \land V2 \in \mathit{decision}[L2,\, B2] \Rightarrow$
$V1 = V2$

Liveness

WORK IN PROGRESS

For any learner L, ballot b and quorum Q of safe acceptors trusted by L such that

0. No phase $1a$ messages (a) have been or (b) ever will be sent for any ballot number greater than b.
1. No $2a$ messages were sent in any round
2. The ballot b leader eventually sends a $1a$ message for ballot b.
3. Each acceptor in Q eventually receives the $1a$ message of ballot b and responds to it by sending a $1b$ message.
4. (a) Each acceptor in Q eventually receives a $1b$ message of ballot b from themself and every other acceptor of Q, and
   (b) sends $2a$ containing the quorum of $1b$ messages to every learner which live-trusts the quorum Q. In particular, the $2a$ messages are sent to the learner L.
5. The learner L receives all $2a$ messages of ballot b addressed to it.
6. Learner L eventually executes its decision action for ballot b if it has a chance to do so.

then some value is eventually chosen by the learner L.

THEOREM $\mathit{Liveness} \triangleq$
$\mathit{Spec} \Rightarrow$
$\forall\, L \in \mathit{Learner} : \forall\, b \in \mathit{Ballot},\, Q \in \mathit{ByzQuorum} :$
$Q \subseteq \mathit{SafeAcceptor} \land$
$[\mathit{lr} \mapsto L,\, q \mapsto Q] \in \mathit{TrustLive} \Rightarrow$
(
(
$(0a)$
$\land \forall\, m \in \mathit{msgs} : m.\mathit{type} = \text{“1a”} \Rightarrow m.\mathit{bal} < b$
$(0b)$
$\land \forall\, c \in \mathit{Ballot} : c > b \Rightarrow \Box[\neg \mathit{Send1a}(c)]_{\mathit{vars}}$
(2)
$\land \mathrm{WF}_{\mathit{vars}}(\mathit{Send1a}(b))$
(3)

$$
\begin{array}{l}
\quad \land \forall\, m \in \mathit{Message} : \\
\qquad B(m,\, b) \Rightarrow \\
\qquad \forall\, a \in Q : \mathrm{WF}_{\mathit{vars}}(\mathit{Process1a}(a,\, m)) \\
\quad (4a) \\
\quad \land \forall\, m \in \mathit{Message} : \\
\qquad B(m,\, b) \Rightarrow \\
\qquad \forall\, a \in Q : \mathrm{WF}_{\mathit{vars}}(\mathit{Process1b}(a,\, m)) \\
\quad (4b) \\
\quad \land \forall\, a \in Q : \mathrm{WF}_{\mathit{vars}}(\mathit{Process1bLearnerLoop}(a)) \\
\quad (5) \\
\quad \land \forall\, m \in \mathit{Message} : \\
\qquad B(m,\, b) \Rightarrow \mathrm{WF}_{\mathit{vars}}(\mathit{LearnerRecv}(L,\, m)) \\
\quad (6) \\
\quad \land \mathrm{WF}_{\mathit{vars}}(\exists\, v \in \mathit{Value} : \mathit{LearnerDecide}(L,\, b,\, v)) \\
)\ \leadsto \\
(\exists\, BB \in \mathit{Ballot} : \mathit{decision}[L,\, BB] \neq \{\}) \\
)
\end{array}
$$

CONSTANTS $bb,\ LL,\ QQ$

$$
\begin{array}{l}
\mathit{CSpec} \triangleq \\
\quad \land \mathit{Init} \\
\quad \land \Box[\mathit{Next} \land \forall\, c \in \mathit{Ballot} : c > bb \Rightarrow \neg \mathit{Send1a}(c)]_{\mathit{vars}} \\
\quad \land \mathrm{WF}_{\mathit{vars}}(\mathit{Send1a}(bb)) \\
\quad \land \forall\, m \in \mathit{Message} : \\
\qquad B(m,\, bb) \Rightarrow \\
\qquad \forall\, a \in QQ : \mathrm{WF}_{\mathit{vars}}(\mathit{Process1a}(a,\, m)) \\
\quad \land \forall\, m \in \mathit{Message} : \\
\qquad B(m,\, bb) \Rightarrow \\
\qquad \forall\, a \in QQ : \mathrm{WF}_{\mathit{vars}}(\mathit{Process1b}(a,\, m)) \\
\quad \land \forall\, a \ \in QQ : \mathrm{WF}_{\mathit{vars}}(\mathit{Process1bLearnerLoop}(a)) \\
\quad \land \forall\, m \in \mathit{Message} : \\
\qquad B(m,\, bb) \Rightarrow \mathrm{WF}_{\mathit{vars}}(\mathit{LearnerRecv}(LL,\, m)) \\
\quad \land \mathrm{WF}_{\mathit{vars}}(\exists\, v \in \mathit{Value} : \mathit{LearnerDecide}(LL,\, bb,\, v))
\end{array}
$$

$$
\begin{array}{l}
\mathit{CLiveness} \triangleq \\
\quad (\ \land QQ \subseteq \mathit{SafeAcceptor} \\
\qquad \land [lr \mapsto LL,\ q \mapsto QQ] \in \mathit{TrustLive}) \\
\quad \Rightarrow \\
\quad ((\forall\, m \in \mathit{msgs} : m.\mathit{type} = \text{“}\mathtt{1a}\text{”} \Rightarrow m.\mathit{bal} < bb) \\
\quad \leadsto \\
\quad \exists\, BB \in \mathit{Ballot} : \mathit{decision}[LL,\, BB] \neq \{\})
\end{array}
$$

* Modification History

\ * Last modified *Mon Nov* 07 13:40:50 *CET* 2022 by *karbyshev*
\ * Created *Mon Jul* 25 14:24:03 *CEST* 2022 by *karbyshev*