MODULE *JustInTimePaxos*

EXTENDS *Naturals*, *Sequences*, *FiniteSets*, *TLC*

The set of *Paxos* replicas
CONSTANT *Replicas*

The set of *Paxos* clients
CONSTANT *Clients*

An empty value
CONSTANT *Nil*

Client request/response types+
CONSTANTS
MWriteRequest,
MWriteResponse,
MReadRequest,
MReadResponse

Server request/response types
CONSTANTS
MRepairRequest,
MRepairResponse,
MAbortRequest,
MAbortResponse,
MViewChangeRequest,
MViewChangeResponse,
MStartViewRequest

Replica roles
CONSTANTS
SNormal,
SAborting,
SViewChange

Entry types
CONSTANTS
TValue,
TNoOp

VARIABLE *replicas*

$globalVars \triangleq \langle replicas \rangle$

VARIABLE *messages*

$messageVars \triangleq \langle messages \rangle$

VARIABLE $cTime$

VARIABLE $cViewID$

VARIABLE $cSeqNum$

VARIABLE $cResps$

VARIABLE $cWrites$

VARIABLE $cReads$

$clientVars \triangleq \langle cTime,\ cViewID,\ cSeqNum,\ cResps,\ cWrites,\ cReads \rangle$

VARIABLE $rStatus$

VARIABLE $rLog$

VARIABLE $rViewID$

VARIABLE $rSeqNum$

VARIABLE $rLastView$

VARIABLE $rViewChanges$

VARIABLE $rAbortSeqNum$

VARIABLE $rAbortResps$

$replicaVars \triangleq \langle rStatus,\ rLog,\ rViewID,\ rSeqNum,\ rLastView,\ rViewChanges,\ rAbortSeqNum,\ rAbortResps \rangle$

VARIABLE $transitions$

$vars \triangleq \langle globalVars,\ messageVars,\ clientVars,\ replicaVars,\ transitions \rangle$

Helpers

RECURSIVE $SeqFromSet(\_)$
$SeqFromSet(S) \triangleq$
  IF $S = \{\}$ THEN $\langle\rangle$
  ELSE LET $x \triangleq$ CHOOSE $x \in S$: TRUE
       IN $\langle x \rangle \circ SeqFromSet(S \setminus \{x\})$

$Max(s) \triangleq$ CHOOSE $x \in s : \forall\, y \in s : x \geq y$

$IsQuorum(s) \triangleq Cardinality(s) * 2 \geq Cardinality(Replicas)$

$Quorums \triangleq \{r \in$ SUBSET $Replicas : IsQuorum(r)\}$

$Primary(v) \triangleq replicas[(v\%Len(replicas)) + (\text{IF } v \geq Len(replicas) \text{ THEN } 1 \text{ ELSE } 0)]$

$IsPrimary(r) \triangleq Primary(rViewID[r]) = r$

$Replace(l,\, i,\, x) \triangleq [j \in 1 \,..\, Max(\{Len(l),\, i\}) \mapsto \text{IF } j = i \text{ THEN } x \text{ ELSE } l[j]]$

Messaging helpers

$Sends(ms) \triangleq messages' = messages \cup ms$

$Send(m) \triangleq Sends(\{m\})$

$Replies(req,\, resps) \triangleq messages' = (messages \cup resps) \setminus \{req\}$

$Reply(req,\, resp) \triangleq Replies(req,\, \{resp\})$

$Discard(m) \triangleq messages' = messages \setminus \{m\}$

$Write(c) \triangleq$
- $\wedge\ cTime' = cTime + 1$
- $\wedge\ cSeqNum' = [cSeqNum \text{ EXCEPT } ![c] = cSeqNum[c] + 1]$
- $\wedge\ Sends(\{[src \mapsto c,\ dest \mapsto r,\ type \mapsto MWriteRequest,\ viewID \mapsto cViewID[c],\ seqNum \mapsto cSeqNum'[c],\ timestamp \mapsto cTime'] : r \in Replicas\})$
- $\wedge$ UNCHANGED $\langle globalVars,\, replicaVars,\, cViewID,\, cResps,\, cWrites,\, cReads\rangle$

$Read(c) \triangleq$
- $\wedge\ Sends(\{[src \mapsto c,\ dest \mapsto r,\ type \mapsto MReadRequest,\ viewID \mapsto cViewID[c]] : r \in Replicas\})$
- $\wedge$ UNCHANGED $\langle globalVars,\, replicaVars,\, cTime,\, cSeqNum,\, cResps,\, cWrites,\, cReads\rangle$

$HandleWriteResponse(c,\, r,\, m) \triangleq$
- $\wedge\ \vee\ \wedge\ m.viewID = cViewID[c]$
  - $\wedge$ IF $m.seqNum \notin$ DOMAIN $cResps[c][r]$ THEN
    - $cResps' = [cResps \text{ EXCEPT } ![c] = [cResps[c] \text{ EXCEPT } ![r] = cResps[c][r] \,@@\, (m.seqNum :> m)]]$
    - ELSE
    - $cResps' = [cResps \text{ EXCEPT } ![c] = [cResps[c] \text{ EXCEPT } ![r] = [cResps[c][r] \text{ EXCEPT } ![m.seqNum]$
  - $\wedge$ LET
    - $allResps \triangleq \{cResps[c][r][r1] : r1 \in \{r2 \in Replicas : r2 \in \text{DOMAIN } cResps[c][r]\}\}$
    - $isCommitted \triangleq \{r1.src : r1 \in \{r2 \in allResps : r2.succeeded\}\} \in Quorums$

```
        IN
            ∧ ∨ ∧ isCommitted
                  ∧ cWrites' = [cWrites EXCEPT ![c] = cWrites[c] ∪ {m}]
              ∨ ∧ ¬isCommitted
                  ∧ UNCHANGED ⟨cWrites⟩
              ∧ UNCHANGED ⟨cViewID, cSeqNum⟩
          ∨ ∧ m.viewID > cViewID[c]
            ∧ cViewID'  = [cViewID EXCEPT ![c] = m.viewID]
            ∧ cSeqNum'  = [cSeqNum EXCEPT ![c] = 0]
            ∧ cResps'   = [cResps  EXCEPT ![c]  = [i ∈ Replicas ↦ {}]]
            ∧ UNCHANGED ⟨cWrites⟩
          ∨ ∧ m.viewID < cViewID[c]
            ∧ UNCHANGED ⟨cWrites⟩
      ∧ Discard(m)
      ∧ UNCHANGED ⟨globalVars, replicaVars, cTime, cSeqNum, cReads⟩

HandleReadResponse(c, r, m) ≜
      ∧ ∨ ∧ m.viewID = cViewID[c]
            ∧ cReads' = [cReads EXCEPT ![c] = cReads[c] ∪ {m}]
            ∧ UNCHANGED ⟨cViewID, cSeqNum⟩
          ∨ ∧ m.viewID > cViewID[c]
            ∧ cViewID'  = [cViewID EXCEPT ![c] = m.viewID]
            ∧ cSeqNum'  = [cSeqNum EXCEPT ![c] = 0]
            ∧ UNCHANGED ⟨cReads⟩
          ∨ ∧ m.viewID < cViewID[c]
            ∧ UNCHANGED ⟨cViewID, cSeqNum, cReads⟩
      ∧ Discard(m)
      ∧ UNCHANGED ⟨globalVars, replicaVars, cTime, cSeqNum, cResps, cWrites⟩
```

Server request/response handling

```
Repair(r, c, m) ≜
      ∧ Replies(m, {[src      ↦ r,
                     dest     ↦ d,
                     type     ↦ MRepairRequest,
                     viewID   ↦ rViewID[r],
                     client   ↦ c,
                     seqNum ↦ rSeqNum[r][c] + 1] : d ∈ Replicas})

Abort(r, c, m) ≜
      ∧ IsPrimary(r)
      ∧ rStatus[r]      = SNormal
      ∧ rStatus'        = [rStatus      EXCEPT ![r]    = SAborting]
      ∧ rAbortResps'    = [rAbortResps  EXCEPT ![r]    = [rAbortResps[r] EXCEPT ![c] = {}]]
```

$\land\ rAbortSeqNum' = [rAbortSeqNum \text{ EXCEPT } ![r] = [rAbortSeqNum[r] \text{ EXCEPT } ![c] = m.seqNum]]$
$\land\ Replies(m, \{[src \mapsto r,$
$dest \mapsto d,$
$type \mapsto MAbortRequest,$
$viewID \mapsto rViewID[r],$
$client \mapsto c,$
$seqNum \mapsto m.seqNum] : d \in Replicas\})$

$HandleWriteRequest(r, c, m) \triangleq$
$\land\ rStatus[r] = SNormal$
$\land\ \lor\ \land\ m.viewID = rViewID[r]$
$\land$ LET
$isSequential \triangleq m.seqNum = rSeqNum[r][c] + 1$
$isLinear \triangleq \forall\, i \in \text{DOMAIN } rLog[r] : \forall\, e \in rLog[r][i] : m.timestamp > e.timestamp$
IN
$\lor\ \land\ isSequential$
$\land\ isLinear$
$\land\ rLog' = [rLog \text{ EXCEPT } ![r] = [$
$rLog[r] \text{ EXCEPT } ![c] =$
$Append(rLog[r][c], [type \mapsto TValue,$
$value \mapsto m.value,$
$timestamp \mapsto m.timestamp])]]$
$\land\ rSeqNum' = [rSeqNum \text{ EXCEPT } ![r] = [rSeqNum[r] \text{ EXCEPT } ![c] = m.seqNum]]$
$\land\ Reply(m, [src \mapsto r,$
$dest \mapsto c,$
$type \mapsto MWriteResponse,$
$seqNum \mapsto m.seqNum,$
$viewID \mapsto rViewID[r],$
$succeeded \mapsto \text{TRUE}])$
$\lor\ \land\ \lor \neg isSequential$
$\lor \neg isLinear$
$\land\ \lor\ \land\ IsPrimary(r)$
$\land\ Abort(r, c, m)$
$\lor\ \land \neg IsPrimary(r)$
$\land\ Repair(r, c, m)$
$\land$ UNCHANGED $\langle rLog \rangle$
$\lor\ \land\ m.viewID < rViewID[r]$
$\land\ Reply(m, [src \mapsto r,$
$dest \mapsto c,$
$type \mapsto MWriteResponse,$
$seqNum \mapsto m.seqNum,$
$viewID \mapsto rViewID[r],$
$succeeded \mapsto \text{FALSE}])$
$\land$ UNCHANGED $\langle rLog \rangle$
$\land$ UNCHANGED $\langle globalVars, clientVars, rStatus, rViewID, rLastView, rViewChanges \rangle$

$$
\begin{array}{l}
\mathit{HandleReadRequest}(r,\ c,\ m)\ \triangleq \\
\quad \land\ \mathit{rStatus}[r] = \mathit{SNormal} \\
\quad \land\ \mathit{Len}(\mathit{rLog}[r]) > 0 \\
\quad \land\ \mathit{Reply}(m,\ [\mathit{src} \mapsto r, \\
\qquad\qquad \mathit{dest} \mapsto c, \\
\qquad\qquad \mathit{type} \mapsto \mathit{MReadResponse}, \\
\qquad\qquad \mathit{viewID} \mapsto \mathit{rViewID}[r], \\
\qquad\qquad \mathit{primary} \mapsto \mathit{IsPrimary}(r), \\
\qquad\qquad \mathit{index} \mapsto \mathit{Len}(\mathit{rLog}[r]), \\
\qquad\qquad \mathit{checksum} \mapsto \mathit{rLog}[r][\mathit{Len}(\mathit{rLog}[r])].\mathit{checksum}, \\
\qquad\qquad \mathit{succeeded} \mapsto \text{TRUE}]) \\
\quad \land\ \text{UNCHANGED}\ \langle \mathit{globalVars},\ \mathit{clientVars},\ \mathit{rStatus},\ \mathit{rLog},\ \mathit{rViewID},\ \mathit{rLastView},\ \mathit{rViewChanges} \rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{HandleRepairRequest}(r,\ s,\ m)\ \triangleq \\
\quad \land\ m.\mathit{viewID} = \mathit{rViewID}[r] \\
\quad \land\ \mathit{IsPrimary}(r) \\
\quad \land\ \mathit{rStatus}[r] = \mathit{SNormal} \\
\quad \land\ \lor\ \land\ m.\mathit{seqNum} \leq \mathit{Len}(\mathit{rLog}[r][m.\mathit{client}]) \\
\qquad\quad \land\ \mathit{Reply}(m,\ [\mathit{src} \mapsto r, \\
\qquad\qquad\qquad \mathit{dest} \mapsto s, \\
\qquad\qquad\qquad \mathit{type} \mapsto \mathit{MRepairResponse}, \\
\qquad\qquad\qquad \mathit{viewID} \mapsto \mathit{rViewID}[r], \\
\qquad\qquad\qquad \mathit{client} \mapsto m.\mathit{client}, \\
\qquad\qquad\qquad \mathit{seqNum} \mapsto m.\mathit{seqNum}]) \\
\qquad\quad \land\ \text{UNCHANGED}\ \langle \mathit{rStatus},\ \mathit{rAbortResps},\ \mathit{rAbortSeqNum} \rangle \\
\qquad \lor\ \land\ m.\mathit{seqNum} = \mathit{Len}(\mathit{rLog}[r][m.\mathit{client}]) + 1 \\
\qquad\quad \land\ \mathit{Abort}(r,\ m.\mathit{client},\ m) \\
\quad \land\ \text{UNCHANGED}\ \langle \mathit{globalVars},\ \mathit{clientVars} \rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{HandleRepairResponse}(r,\ s,\ m)\ \triangleq \\
\quad \land\ \mathit{HandleWriteRequest}(r,\ m.\mathit{client},\ [m\ \text{EXCEPT}\ !.\mathit{src} = m.\mathit{client}])
\end{array}
$$

$$
\begin{array}{l}
\mathit{HandleAbortRequest}(r,\ s,\ m)\ \triangleq \\
\quad \land\ m.\mathit{viewID}\ = \mathit{rViewID}[r] \\
\quad \land\ m.\mathit{seqNum} \leq \mathit{Len}(\mathit{rLog}[r][m.\mathit{client}]) + 1 \\
\quad \land\ \mathit{rStatus}[r] \in \{\mathit{SNormal},\ \mathit{SAborting}\} \\
\quad \land\ \mathit{rLog}' = [\mathit{rLog}\ \text{EXCEPT}\ ![r] = [\mathit{rLog}[r]\ \text{EXCEPT}\ ![m.\mathit{client}] = \mathit{Replace}(\mathit{rLog}[r][m.\mathit{client}],\ m.\mathit{seqNum},\ [\mathit{type} \\
\quad \land\ \lor\ \land\ m.\mathit{seqNum} > \mathit{rSeqNum}[r][m.\mathit{client}] \\
\qquad\quad \land\ \mathit{rSeqNum}'\ = [\mathit{rSeqNum}\ \text{EXCEPT}\ ![r] = [\mathit{rSeqNum}[r]\ \text{EXCEPT}\ ![m.\mathit{client}] = m.\mathit{seqNum}]] \\
\qquad \lor\ \land\ m.\mathit{seqNum} \leq \mathit{rSeqNum}[r][m.\mathit{client}] \\
\qquad\quad \land\ \text{UNCHANGED}\ \langle \mathit{rSeqNum} \rangle \\
\quad \land\ \mathit{Replies}(m,\ \{[\mathit{src} \mapsto r, \\
\qquad\qquad\qquad \mathit{dest} \mapsto \mathit{Primary}(\mathit{rViewID}[r]), \\
\qquad\qquad\qquad \mathit{type} \mapsto \mathit{MAbortResponse}, \\
\qquad\qquad\qquad \mathit{viewID} \mapsto \mathit{rViewID}[r], \\
\qquad\qquad\qquad \mathit{seqNum} \mapsto m.\mathit{seqNum}],
\end{array}
$$

$$
\begin{array}{l}
\quad\quad\quad\quad [src \mapsto r,\\
\quad\quad\quad\quad\ dest \mapsto Primary(rViewID[r]),\\
\quad\quad\quad\quad\ type \mapsto MWriteResponse,\\
\quad\quad\quad\quad\ viewID \mapsto rViewID[r],\\
\quad\quad\quad\quad\ seqNum \mapsto m.seqNum,\\
\quad\quad\quad\quad\ succeeded \mapsto \text{FALSE}]\})\\
\quad \land \text{UNCHANGED } \langle globalVars,\ clientVars,\ rStatus,\ rViewID,\ rLastView,\ rViewChanges\rangle
\end{array}
$$

$$
\begin{array}{l}
HandleAbortResponse(r,\ s,\ m) \triangleq\\
\quad \land\ rStatus[r] = SAborting\\
\quad \land\ m.viewID = rViewID[r]\\
\quad \land\ IsPrimary(r)\\
\quad \land\ m.seqNum = rAbortSeqNum[r][m.client]\\
\quad \land\ rAbortResps' = [rAbortResps \text{ EXCEPT } ![r] = [rAbortResps[r] \text{ EXCEPT } ![m.client] = rAbortResps[r][m.cl\\
\quad \land \text{LET } resps \triangleq \{res.src : res \in \{resp \in rAbortResps'[r][m.client] :\\
\quad\quad\quad\quad\quad\quad\quad\quad \land\ resp.viewID\ = rViewID[r]\\
\quad\quad\quad\quad\quad\quad\quad\quad \land\ resp.seqNum = rAbortSeqNum[r][m.client]\}\}\\
\quad\quad\quad isQuorum \triangleq\ r \in resps \land resps \in Quorums\\
\quad\ \text{IN}\\
\quad\quad \lor\ \land\ isQuorum\\
\quad\quad\quad\ \land\ rStatus' = [rStatus \text{ EXCEPT } ![r] = [rStatus[r] \text{ EXCEPT } ![m.client] = SNormal]]\\
\quad\quad \lor\ \land\ \neg isQuorum\\
\quad\quad\quad\ \land \text{UNCHANGED } \langle rStatus\rangle\\
\quad \land \text{UNCHANGED } \langle globalVars,\ clientVars\rangle
\end{array}
$$

$$
\begin{array}{l}
ChangeView(r) \triangleq\\
\quad \land\ Sends(\{[src \mapsto r,\\
\quad\quad\quad\quad\quad dest \mapsto d,\\
\quad\quad\quad\quad\quad type \mapsto MViewChangeRequest,\\
\quad\quad\quad\quad\quad viewID \mapsto rViewID[r] + 1] : d \in Replicas\})\\
\quad \land \text{UNCHANGED } \langle globalVars,\ clientVars,\ replicaVars\rangle
\end{array}
$$

$$
\begin{array}{l}
HandleViewChangeRequest(r,\ s,\ m) \triangleq\\
\quad \land\ rViewID[r] < m.viewID\\
\quad \land\ rViewID' = [rViewID \text{ EXCEPT } ![r] = m.viewID]\\
\quad \land\ rStatus' = [rStatus \text{ EXCEPT } ![r] = SViewChange]\\
\quad \land\ rViewChanges' = [rViewChanges \text{ EXCEPT } ![r] = \{\}]\\
\quad \land\ Reply(m, [src \mapsto r,\\
\quad\quad\quad\quad\quad dest \mapsto Primary(m.viewID),\\
\quad\quad\quad\quad\quad type \mapsto MViewChangeResponse,\\
\quad\quad\quad\quad\quad viewID \mapsto m.viewID,\\
\quad\quad\quad\quad\quad lastViewID \mapsto rLastView[r],\\
\quad\quad\quad\quad\quad logs \mapsto rLog[r]])\\
\quad \land \text{UNCHANGED } \langle globalVars,\ clientVars,\ rLog,\ rSeqNum,\ rAbortSeqNum,\ rAbortResps,\ rLastView\rangle
\end{array}
$$

$$
HandleViewChangeResponse(r,\ s,\ m) \triangleq
$$

```
    ∧ IsPrimary(r)
    ∧ rViewID[r]      = m.viewID
    ∧ rStatus[r]      = SViewChange
    ∧ rViewChanges' = [rViewChanges EXCEPT ![r] = rViewChanges[r] ∪ {m}]
    ∧ LET viewChanges   ≜ {v ∈ rViewChanges'[r][m.client] : ∧ v.viewID = rViewID[r]}
          viewSources   ≜ {v.src : v ∈ viewChanges}
          isQuorum      ≜ r ∈ viewSources ∧ viewSources ∈ Quorums
          lastViews     ≜ {v.lastViewID : v ∈ viewChanges}
          lastView      ≜ (CHOOSE v1 ∈ lastViews : ∀ v2 ∈ lastViews : v2 ≤ v1)
          viewLogs      ≜ [c ∈ Clients ↦ {v1.logs[c] : v1 ∈ {v2 ∈ viewChanges : v2.lastView = lastView}
          mergeEnts(es) ≜
              IF es = {} ∨ ∃ e ∈ es : r.type = TNoOp THEN
                  [type ↦ TNoOp]
               ELSE
                  CHOOSE e ∈ es : e.type ≠ TNoOp
          range(ls)     ≜ Max({Len(l) : l ∈ ls})
          entries(ls, i) ≜ {l[i] : l ∈ {k ∈ ls : i ≤ Len(k)}}
          mergeLogs(ls) ≜ [i ∈ 1 .. range(ls) ↦ mergeEnts(entries(ls, i))]
      IN
        ∨ ∧ isQuorum
          ∧ Replies(m, {[src      ↦ r,
                         dest     ↦ d,
                         type     ↦ MStartViewRequest,
                         viewID   ↦ rViewID[r],
                         logs     ↦ [c ∈ Clients ↦ mergeLogs(viewLogs[c])]] : d ∈ Replicas})
        ∨ ∧ ¬isQuorum
          ∧ Discard(m)
    ∧ UNCHANGED ⟨globalVars, clientVars, rStatus, rViewID, rLog, rSeqNum, rAbortSeqNum, rAbortResps, r

HandleStartViewRequest(r, s, m) ≜
    ∧ ∨ rViewID[r] < m.viewID
      ∨ ∧ rViewID[r] = m.viewID
        ∧ rStatus[r]  = SViewChange
    ∧ rLog'       = [rLog       EXCEPT ![r]  = m.log]
    ∧ rStatus'    = [rStatus    EXCEPT ![r]  = SNormal]
    ∧ rViewID'    = [rViewID    EXCEPT ![r] = m.viewID]
    ∧ rLastView' = [rLastView EXCEPT ![r] = m.viewID]
    ∧ Discard(m)
    ∧ UNCHANGED ⟨globalVars, clientVars, rViewChanges⟩
```

```
InitMessageVars ≜
    ∧ messages = {}
```

$$
\begin{array}{l}
\mathit{InitClientVars} \triangleq \\
\quad \land\ \mathit{cTime} = 0 \\
\quad \land\ \mathit{cViewID} = [c \in \mathit{Clients} \mapsto 1] \\
\quad \land\ \mathit{cSeqNum} = [c \in \mathit{Clients} \mapsto 0] \\
\quad \land\ \mathit{cResps} = [c \in \mathit{Clients} \mapsto [r \in \mathit{Replicas} \mapsto [s \in \{\} \mapsto [\mathit{index} \mapsto 0,\ \mathit{checksum} \mapsto \mathit{Nil}]]]] \\
\quad \land\ \mathit{cWrites} = [c \in \mathit{Clients} \mapsto \{\}] \\
\quad \land\ \mathit{cReads} = [c \in \mathit{Clients} \mapsto \{\}]
\end{array}
$$

$$
\begin{array}{l}
\mathit{InitReplicaVars} \triangleq \\
\quad \land\ \mathit{replicas} = \mathit{SeqFromSet}(\mathit{Replicas}) \\
\quad \land\ \mathit{rStatus} = [r \in \mathit{Replicas} \mapsto \mathit{SNormal}] \\
\quad \land\ \mathit{rLog} = [r \in \mathit{Replicas} \mapsto [c \in \mathit{Clients} \mapsto \langle\rangle]] \\
\quad \land\ \mathit{rSeqNum} = [r \in \mathit{Replicas} \mapsto [c \in \mathit{Clients} \mapsto 0]] \\
\quad \land\ \mathit{rAbortSeqNum} = [r \in \mathit{Replicas} \mapsto [c \in \mathit{Clients} \mapsto 0]] \\
\quad \land\ \mathit{rAbortResps} = [r \in \mathit{Replicas} \mapsto [c \in \mathit{Clients} \mapsto \{\}]] \\
\quad \land\ \mathit{rViewID} = [r \in \mathit{Replicas} \mapsto 1] \\
\quad \land\ \mathit{rLastView} = [r \in \mathit{Replicas} \mapsto 1] \\
\quad \land\ \mathit{rViewChanges} = [r \in \mathit{Replicas} \mapsto \{\}]
\end{array}
$$

$$
\begin{array}{l}
\mathit{Init} \triangleq \\
\quad \land\ \mathit{InitMessageVars} \\
\quad \land\ \mathit{InitClientVars} \\
\quad \land\ \mathit{InitReplicaVars} \\
\quad \land\ \mathit{transitions} = 0
\end{array}
$$

The type invariant checks that no read ever reads a different value than a previous write

$$\mathit{Inv} \triangleq \text{TRUE} \quad \mathit{TODO}$$

$$\mathit{Transition} \triangleq \mathit{transitions}' = \mathit{transitions} + 1$$

$$
\begin{array}{l}
\mathit{Next} \triangleq \\
\quad \lor\ \exists\, c \in \mathit{Clients} : \\
\qquad \land\ \mathit{Write}(c) \\
\qquad \land\ \mathit{Transition} \\
\quad \lor\ \exists\, c \in \mathit{Clients} : \\
\qquad \land\ \mathit{Read}(c) \\
\qquad \land\ \mathit{Transition} \\
\quad \lor\ \exists\, r \in \mathit{Replicas} : \\
\qquad \land\ \mathit{ChangeView}(r) \\
\qquad \land\ \mathit{Transition} \\
\quad \lor\ \exists\, m \in \mathit{messages} : \\
\qquad \land\ m.\mathit{type} = \mathit{MWriteRequest} \\
\qquad \land\ \mathit{HandleWriteRequest}(m.\mathit{dest},\ m.\mathit{src},\ m) \\
\qquad \land\ \mathit{Transition} \\
\quad \lor\ \exists\, m \in \mathit{messages} :
\end{array}
$$

$\land\ m.type = MWriteResponse$
$\land\ HandleWriteResponse(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MReadRequest$
$\land\ HandleReadRequest(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MReadResponse$
$\land\ HandleReadResponse(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MRepairRequest$
$\land\ HandleRepairRequest(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MRepairResponse$
$\land\ HandleRepairResponse(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MAbortRequest$
$\land\ HandleAbortRequest(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MAbortResponse$
$\land\ HandleAbortResponse(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MViewChangeRequest$
$\land\ HandleViewChangeRequest(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MViewChangeResponse$
$\land\ HandleViewChangeResponse(m.dest,\ m.src,\ m)$
$\land\ Transition$
$\lor\ \exists\, m \in messages :$
$\land\ m.type = MStartViewRequest$
$\land\ HandleStartViewRequest(m.dest,\ m.src,\ m)$
$\land\ Transition$

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

\ * Modification History
\ * Last modified *Tue Sep* 22 04:25:22 *PDT* 2020 by *jordanhalterman*
\ * Created *Fri Sep* 18 22:45:21 *PDT* 2020 by *jordanhalterman*