```
──────────────────── MODULE ZabWithQTest2 ────────────────────
EXTENDS Zab

──────────────────────────────────────────────────────────────
  constants that uniquely used for constraining state space in model checking
CONSTANTS MaxElectionNum, MaxTotalRestartNum, MaxTransactionNum

──────────────────────────────────────────────────────────────
  variables that uniquely used for constraining state space in model checking
VARIABLES electionNum,      the round of leader election, not equal to Maximum{currentEpoch[i] : i ∈ Server},
                            because currentEpoch will increase only when follower receives NEWEPOCH,
                            and it is common that some round of election ends without leader broadcasting NEWEPOCH
                            or follower receiving NEWEPOCH.
          totalRestartNum   the number of restart from all servers, also as a global variable.

testVars  ≜ ⟨electionNum, totalRestartNum⟩

varsT ≜ ⟨vars, testVars⟩

──────────────────────────────────────────────────────────────
InitT ≜ ∧ Init
        ∧ electionNum     = 0
        ∧ totalRestartNum = 0

──────────────────────────────────────────────────────────────
ElectionT(i, Q) ≜   test restrictions
        ∧ electionNum < MaxElectionNum
        ∧ Election(i, Q)
        ∧ electionNum′ = electionNum + 1

InitialElectionT(i, Q) ≜
        ∧ ∀ s ∈ Server : state[s] = Follower ∧ leaderOracle[s] = NullPoint
        ∧ ElectionT(i, Q)
        ∧ UNCHANGED ⟨currentEpoch, history, commitIndex, currentCounter,
                     sendCounter, recoveryVars, proposalMsgsLog, totalRestartNum⟩

LeaderTimeoutT(i, j) ≜
        ∧ state[i] ≠ Follower
        ∧ j ≠ i
        ∧ j ∈ cluster[i]
        ∧ LET newCluster ≜ cluster[i] \ {j}
          IN   ∧ ∨ ∧ newCluster ∈ Quorums
                   ∧ cluster′ = [cluster EXCEPT ![i] = newCluster]
                   ∧ clean(i, j)
                   ∧ UNCHANGED ⟨state, cepochRecv, ackeRecv, ackldRecv, ackIndex,
                                committedIndex, initialHistory, tempMaxEpoch, tempMaxLastEpoch,
                                tempInitialHistory, leaderOracle, leaderEpoch, cepochSent, electionNum⟩
```

```
                  ∨ ∧ newCluster ∉ Quorums
                    ∧ ∃ Q ∈ Quorums : ∧ i ∈ Q
                                      ∧ ∃ v ∈ Q : ElectionT(v, Q)
          ∧ UNCHANGED ⟨currentEpoch, history, commitIndex, currentCounter, sendCounter,
                       recoveryVars, proposalMsgsLog, totalRestartNum⟩

FollowerTimeoutT(i) ≜
          ∧ state[i] = Follower
          ∧ leaderOracle[i] ≠ NullPoint
          ∧ ∃ Q ∈ Quorums : ∧ i ∈ Q
                            ∧ ∃ v ∈ Q : ElectionT(v, Q)
          ∧ UNCHANGED ⟨currentEpoch, history, commitIndex, currentCounter, sendCounter,
                        recoveryVars, proposalMsgsLog, totalRestartNum⟩

|------------------------------------------------------------------------------|
RestartT(i) ≜   test restrictions
            ∧ totalRestartNum < MaxTotalRestartNum
            ∧ totalRestartNum′ = totalRestartNum + 1
            ∧ Restart(i)
            ∧ UNCHANGED electionNum

RecoveryAfterRestartT(i) ≜   test restrictions
          ∧ totalRestartNum < MaxTotalRestartNum
          ∧ RecoveryAfterRestart(i)
          ∧ UNCHANGED testVars

HandleRecoveryRequestT(i, j) ≜ ∧ HandleRecoveryRequest(i, j)
                               ∧ UNCHANGED testVars

HandleRecoveryResponseT(i, j) ≜ ∧ HandleRecoveryResponse(i, j)
                                ∧ UNCHANGED testVars

FindClusterT(i) ≜ ∧ FindCluster(i)
                  ∧ UNCHANGED testVars
|------------------------------------------------------------------------------|
FollowerDiscovery1T(i) ≜ ∧ FollowerDiscovery1(i)
                         ∧ UNCHANGED testVars

LeaderHandleCEPOCHT(i, j) ≜ ∧ LeaderHandleCEPOCH(i, j)
                            ∧ UNCHANGED testVars

LeaderDiscovery1T(i) ≜ ∧ LeaderDiscovery1(i)
                       ∧ UNCHANGED testVars

FollowerDiscovery2T(i, j) ≜ ∧ FollowerDiscovery2(i, j)
                            ∧ UNCHANGED testVars

LeaderHandleACKET(i, j) ≜ ∧ LeaderHandleACKE(i, j)
```

$$\land \textsc{unchanged}\ \mathit{testVars}$$

$$\mathit{LeaderDiscovery2Sync1T}(i) \triangleq \begin{array}{l} \land \mathit{LeaderDiscovery2Sync1}(i) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{FollowerSync1T}(i, j) \triangleq \begin{array}{l} \land \mathit{FollowerSync1}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderHandleACKLDT}(i, j) \triangleq \begin{array}{l} \land \mathit{LeaderHandleACKLD}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderSync2T}(i) \triangleq \begin{array}{l} \land \mathit{LeaderSync2}(i) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{FollowerSync2T}(i, j) \triangleq \begin{array}{l} \land \mathit{FollowerSync2}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$\mathit{ClientRequestT}(i, v) \triangleq$ test restrictions

$$\begin{array}{l} \land \mathit{Len}(\mathit{history}[i]) < \mathit{MaxTransactionNum} \\ \land \mathit{ClientRequest}(i, v) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderBroadcast1T}(i) \triangleq \begin{array}{l} \land \mathit{LeaderBroadcast1}(i) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{FollowerBroadcast1T}(i, j) \triangleq \begin{array}{l} \land \mathit{FollowerBroadcast1}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderHandleACKT}(i, j) \triangleq \begin{array}{l} \land \mathit{LeaderHandleACK}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderAdvanceCommitT}(i) \triangleq \begin{array}{l} \land \mathit{LeaderAdvanceCommit}(i) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderBroadcast2T}(i) \triangleq \begin{array}{l} \land \mathit{LeaderBroadcast2}(i) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{FollowerBroadcast2T}(i, j) \triangleq \begin{array}{l} \land \mathit{FollowerBroadcast2}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderHandleCEPOCHinPhase3T}(i, j) \triangleq \begin{array}{l} \land \mathit{LeaderHandleCEPOCHinPhase3}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{LeaderHandleACKLDinPhase3T}(i, j) \triangleq \begin{array}{l} \land \mathit{LeaderHandleACKLDinPhase3}(i, j) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$\mathit{BecomeFollowerT}(i) \triangleq \begin{array}{l} \land \mathit{BecomeFollower}(i) \\ \land \textsc{unchanged}\ \mathit{testVars} \end{array}$$

$$
\begin{array}{ll}
DiscardStaleMessageT(i) \triangleq & \land\ DiscardStaleMessage(i) \\
 & \land\ \text{UNCHANGED}\ \ testVars
\end{array}
$$

Defines how the variables may transition.

$$
\begin{array}{lll}
NextT \triangleq & & \\
\quad \lor\ \exists\, i \in Server,\ Q \in Quorums : & InitialElectionT(i,\ Q) & \\
\quad \lor\ \exists\, i \in Server : & RestartT(i) & \\
\quad \lor\ \exists\, i \in Server : & RecoveryAfterRestartT(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & HandleRecoveryRequestT(i,\ j) & \\
\quad \lor\ \exists\, i,\ j \in Server : & HandleRecoveryResponseT(i,\ j) & \\
\quad \lor\ \exists\, i,\ j \in Server : & FindClusterT(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & LeaderTimeoutT(i,\ j) & \\
\quad \lor\ \exists\, i \in Server : & FollowerTimeoutT(i) & \\
\quad \lor\ \exists\, i \in Server : & FollowerDiscovery1T(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & LeaderHandleCEPOCHT(i,\ j) & \\
\quad \lor\ \exists\, i \in Server : & LeaderDiscovery1T(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & FollowerDiscovery2T(i,\ j) & \\
\quad \lor\ \exists\, i,\ j \in Server : & LeaderHandleACKET(i,\ j) & \\
\quad \lor\ \exists\, i \in Server : & LeaderDiscovery2Sync1T(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & FollowerSync1T(i,\ j) & \\
\quad \lor\ \exists\, i,\ j \in Server : & LeaderHandleACKLDT(i,\ j) & \\
\quad \lor\ \exists\, i \in Server : & LeaderSync2T(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & FollowerSync2T(i,\ j) & \\
\quad \lor\ \exists\, i \in Server,\ v \in Value : & ClientRequestT(i,\ v) & \\
\quad \lor\ \exists\, i \in Server : & LeaderBroadcast1T(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & FollowerBroadcast1T(i,\ j) & \\
\quad \lor\ \exists\, i,\ j \in Server : & LeaderHandleACKT(i,\ j) & \\
\quad \lor\ \exists\, i \in Server : & LeaderAdvanceCommitT(i) & \\
\quad \lor\ \exists\, i \in Server : & LeaderBroadcast2T(i) & \\
\quad \lor\ \exists\, i,\ j \in Server : & FollowerBroadcast2T(i,\ j) & \\
\quad \lor\ \exists\, i,\ j \in Server : & LeaderHandleCEPOCHinPhase3T(i,\ j) & \\
\quad \lor\ \exists\, i,\ j \in Server : & LeaderHandleACKLDinPhase3T(i,\ j) & \\
\quad \lor\ \exists\, i \in Server : & DiscardStaleMessageT(i) & \\
\quad \lor\ \exists\, i \in Server : & BecomeFollowerT(i) &
\end{array}
$$

$$SpecT \triangleq InitT \land \Box[NextT]_{varsT}$$

\ * Modification History

\ * Last modified *Mon* May 17 22:00:38 *CST* 2021 by Dell

\ * Created *Mon* May 17 17:20:01 *CST* 2021 by Dell