MODULE *ZabWithFLETest*

EXTENDS *ZabWithFLE*

---

constants that uniquely used for constraining state space in model checking

CONSTANTS *MaxTotalTimeoutNum*, *MaxTransactionNum*

---

variables that uniquely used for constraining state space in model checking

VARIABLES *totalTimeoutNum* the number of timeout from all servers, as a global variable.

$testVars \triangleq \langle totalTimeoutNum \rangle$

$varsT \triangleq \langle vars, testVars \rangle$

$InitT \triangleq \land InitZ$
$\quad \land totalTimeoutNum = 0$

$FLEReceiveNotmsgT(i, j) \triangleq \land FLEReceiveNotmsg(i, j)$
$\quad \land$ UNCHANGED $testVars$

$FLENotmsgTimeoutT(i) \triangleq \land FLENotmsgTimeout(i)$
$\quad \land$ UNCHANGED $testVars$

$FLEHandleNotmsgT(i) \triangleq \land FLEHandleNotmsg(i)$
$\quad \land$ UNCHANGED $testVars$

$FLEWaitNewNotmsgT(i) \triangleq \land FLEWaitNewNotmsg(i)$
$\quad \land$ UNCHANGED $testVars$

$FLEWaitNewNotmsgEndT(i) \triangleq \land FLEWaitNewNotmsgEnd(i)$
$\quad \land$ UNCHANGED $testVars$

$FollowerTimoutT(i) \triangleq$ test restrictions
$\quad \land totalTimeoutNum < MaxTotalTimeoutNum$
$\quad \land totalTimeoutNum' = totalTimeoutNum + 1$
$\quad \land FollowerTimout(i)$

$LeaderTimeoutT(i) \triangleq$ test restrictions
$\quad \land totalTimeoutNum < MaxTotalTimeoutNum$
$\quad \land totalTimeoutNum' = totalTimeoutNum + 1$
$\quad \land LeaderTimeout(i)$

$TimeoutT(i, j) \triangleq$ test restrictions
$\quad \land totalTimeoutNum < MaxTotalTimeoutNum$
$\quad \land totalTimeoutNum' = totalTimeoutNum + 1$
$\quad \land Timeout(i, j)$

$EstablishConnectionT(i, j) \triangleq \land EstablishConnection(i, j)$
$\quad \land$ UNCHANGED $testVars$

$FollowerSendFOLLOWERINFOT(i) \triangleq \land FollowerSendFOLLOWERINFO(i)$
$\land$ UNCHANGED $testVars$

$LeaderHandleFOLLOWERINFOT(i, j) \triangleq \land LeaderHandleFOLLOWERINFO(i, j)$
$\land$ UNCHANGED $testVars$

$LeaderDiscovery1T(i) \triangleq \land LeaderDiscovery1(i)$
$\land$ UNCHANGED $testVars$

$FollowerHandleLEADERINFOT(i, j) \triangleq \land FollowerHandleLEADERINFO(i, j)$
$\land$ UNCHANGED $testVars$

$LeaderHandleACKEPOCHT(i, j) \triangleq \land LeaderHandleACKEPOCH(i, j)$
$\land$ UNCHANGED $testVars$

$LeaderDiscovery2T(i) \triangleq \land LeaderDiscovery2(i)$
$\land$ UNCHANGED $testVars$

$RECOVERYSYNCT(i, j) \triangleq \land RECOVERYSYNC(i, j)$
$\land$ UNCHANGED $testVars$

$FollowerHandleNEWLEADERT(i, j) \triangleq \land FollowerHandleNEWLEADER(i, j)$
$\land$ UNCHANGED $testVars$

$LeaderHandleACKLDT(i, j) \triangleq \land LeaderHandleACKLD(i, j)$
$\land$ UNCHANGED $testVars$

$LeaderSync2T(i) \triangleq \land LeaderSync2(i)$
$\land$ UNCHANGED $testVars$

$FollowerHandleUPTODATET(i, j) \triangleq \land FollowerHandleUPTODATE(i, j)$
$\land$ UNCHANGED $testVars$

$ClientRequestT(i, v) \triangleq$ test restrictions
$\land Len(history[i]) < MaxTransactionNum$
$\land ClientRequest(i, v)$
$\land$ UNCHANGED $testVars$

$LeaderBroadcast1T(i) \triangleq \land LeaderBroadcast1(i)$
$\land$ UNCHANGED $testVars$

$FollowerHandlePROPOSALT(i, j) \triangleq \land FollowerHandlePROPOSAL(i, j)$
$\land$ UNCHANGED $testVars$

$LeaderHandleACKT(i, j) \triangleq \land LeaderHandleACK(i, j)$
$\land$ UNCHANGED $testVars$

$LeaderAdvanceCommitT(i) \triangleq \land LeaderAdvanceCommit(i)$
$\land$ UNCHANGED $testVars$

$LeaderBroadcast2T(i) \triangleq \land\ LeaderBroadcast2(i)$
$\qquad\qquad\qquad\qquad\land\ \text{UNCHANGED}\ testVars$

$FollowerHandleCOMMITT(i, j) \triangleq \land\ FollowerHandleCOMMIT(i, j)$
$\qquad\qquad\qquad\qquad\land\ \text{UNCHANGED}\ testVars$

$FilterNonexistentMessageT(i) \triangleq \land\ FilterNonexistentMessage(i)$
$\qquad\qquad\qquad\qquad\land\ \text{UNCHANGED}\ testVars$

$NextT \triangleq$

FLE modlue
$\lor \exists\, i, j \in Server : FLEReceiveNotmsgT(i, j)$
$\lor \exists\, i \in Server : \quad FLENotmsgTimeoutT(i)$
$\lor \exists\, i \in Server : \quad FLEHandleNotmsgT(i)$
$\lor \exists\, i \in Server : \quad FLEWaitNewNotmsgT(i)$
$\lor \exists\, i \in Server : \quad FLEWaitNewNotmsgEndT(i)$
Some conditions like failure, network delay
$\lor \exists\, i \in Server : \quad FollowerTimoutT(i)$
$\lor \exists\, i \in Server : \quad LeaderTimeoutT(i)$
$\lor \exists\, i, j \in Server : TimeoutT(i, j)$
Zab module - Discovery and Synchronization part
$\lor \exists\, i, j \in Server : EstablishConnectionT(i, j)$
$\lor \exists\, i \in Server : \quad FollowerSendFOLLOWERINFOT(i)$
$\lor \exists\, i, j \in Server : LeaderHandleFOLLOWERINFOT(i, j)$
$\lor \exists\, i \in Server : \quad LeaderDiscovery1T(i)$
$\lor \exists\, i, j \in Server : FollowerHandleLEADERINFOT(i, j)$
$\lor \exists\, i, j \in Server : LeaderHandleACKEPOCHT(i, j)$
$\lor \exists\, i \in Server : \quad LeaderDiscovery2T(i)$
$\lor \exists\, i, j \in Server : RECOVERYSYNCT(i, j)$
$\lor \exists\, i, j \in Server : FollowerHandleNEWLEADERT(i, j)$
$\lor \exists\, i, j \in Server : LeaderHandleACKLDT(i, j)$
$\lor \exists\, i \in Server : \quad LeaderSync2T(i)$
$\lor \exists\, i, j \in Server : FollowerHandleUPTODATET(i, j)$
Zab module - Broadcast part
$\lor \exists\, i \in Server, v \in Value : ClientRequestT(i, v)$
$\lor \exists\, i \in Server : \quad LeaderBroadcast1T(i)$
$\lor \exists\, i, j \in Server : FollowerHandlePROPOSALT(i, j)$
$\lor \exists\, i, j \in Server : LeaderHandleACKT(i, j)$
$\lor \exists\, i \in Server : \quad LeaderAdvanceCommitT(i)$
$\lor \exists\, i \in Server : \quad LeaderBroadcast2T(i)$
$\lor \exists\, i, j \in Server : FollowerHandleCOMMITT(i, j)$
An action used to judge whether there are redundant messages in network
$\lor \exists\, i \in Server : \quad FilterNonexistentMessageT(i)$

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

\ * Modification History
\ * Last modified *Wed Jul* 14 18:46:35 *CST* 2021 by Dell
\ * Created *Wed Jul* 14 15:36:39 *CST* 2021 by Dell