MODULE *StateAWSet*

EXTENDS *AWSet*
CONSTANTS $Read(\_)$

VARIABLES
- $aset$, $aset[r]$: the set of active elements maintained by $r \in Replica$
- $tset$, $tset[r]$: the set of tombstone elements maintained by $r \in Replica$

variables for network:
- $incoming$, $incoming[r]$: incoming channel at replica $r \in Replica$
- $lmsg$, $lmsg[r]$: the last message delivered at $r \in Replica$ to the upper-layer protocol

variables for SEC:
- $uset$, $uset[r]$: the set of updates seen by replica $r \in Replica$
- $uincoming$ $uincoming[r]$: incoming channel for broadcasting/delivering updates at $r \in Replica$

$nVars \triangleq \langle incoming, lmsg \rangle$
$secVars \triangleq \langle uset, uincoming \rangle$
$vars \triangleq \langle aset, tset, seq, nVars, secVars \rangle$

$Msg \triangleq [aid : Aid, A : \text{SUBSET } Element, T : \text{SUBSET } Element]$
$Network \triangleq \text{INSTANCE } BasicNetwork$ WITH $incoming \leftarrow incoming, lmsg \leftarrow lmsg$

$ReadStateAWSet(r) \triangleq \{ele.d : ele \in aset[r]\}$ read the state of $r \in Replica$
$SEC \triangleq \text{INSTANCE } StateSEC$ WITH $uset \leftarrow uset, uincoming \leftarrow uincoming$

$TypeOK \triangleq$
- $\land\ aset \in [Replica \to \text{SUBSET } Element]$
- $\land\ tset \in [Replica \to \text{SUBSET } Element]$
- $\land\ IntTypeOK$
- $\land\ Network!SMTypeOK$
- $\land\ SEC!SECTypeOK$

$Init \triangleq$
- $\land\ aset = [r \in Replica \mapsto \{\}]$
- $\land\ tset = [r \in Replica \mapsto \{\}]$
- $\land\ IntInit$
- $\land\ Network!BNInit$
- $\land\ SEC!StateSECInit$

1
1
1
1
1
1
1
1
1

$Add(d,\ r) \triangleq$ $r \in Replica$ adds $d \in Data$
$\quad \land\ aset' = [aset \text{ EXCEPT } ![r] = @ \cup \{[aid \mapsto [r \mapsto r,\ seq \mapsto seq[r]],\ d \mapsto d]\}]$
$\quad \land\ IntDo(r)$
$\quad \land\ SEC!StateSECDo(r)$
$\quad \land\ \text{UNCHANGED } \langle tset,\ nVars \rangle$

$Remove(d,\ r) \triangleq$ $r \in Replica$ removes $d \in Data$
$\quad \land\ \text{LET } E \triangleq \{ele \in aset[r] : ele.d = d\}$ E may be empty
$\quad\quad \text{IN} \quad \land\ aset' = [aset \text{ EXCEPT } ![r] = @ \setminus E]$
$\quad\quad\quad\quad \land\ tset' = [tset \text{ EXCEPT } ![r] = @ \cup E]$
$\quad \land\ IntDo(r)$
$\quad \land\ SEC!StateSECDo(r)$
$\quad \land\ \text{UNCHANGED } \langle nVars \rangle$

$Do(r) \triangleq$ We ignore $ReadStateAWSet(r)$ since it does not modify states.
$\quad \exists\, a \in Data : Add(a,\ r) \lor Remove(a,\ r)$

$Send(r) \triangleq$ $r \in Replica$ sends a message
$\quad \land\ Network!BNBroadcast(r,\ [aid \mapsto [r \mapsto r,\ seq \mapsto seq[r]],$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad A \mapsto aset[r],\ T \mapsto tset[r]])$
$\quad \land\ IntSend(r)$
$\quad \land\ SEC!StateSECSend(r)$
$\quad \land\ \text{UNCHANGED } \langle aset,\ tset \rangle$

$Deliver(r) \triangleq$ $r \in Replica$ delivers a message ($lmsg'[r]$)
$\quad \land\ IntDeliver(r)$
$\quad \land\ Network!BNDeliver(r)$
$\quad \land\ SEC!StateSECDeliver(r,\ lmsg'[r].aid)$
$\quad \land\ tset' = [tset \text{ EXCEPT } ![r] = @ \cup lmsg'[r].T]$
$\quad \land\ aset' = [aset \text{ EXCEPT } ![r] = (@ \cup lmsg'[r].A) \setminus tset'[r]]$
$\quad \land\ \text{UNCHANGED } \langle\rangle$

$Next \triangleq \exists\, r \in Replica : Do(r) \lor Send(r) \lor Deliver(r)$

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

* Modification History
* Last modified *Wed Jun* 26 15:12:53 *CST* 2019 by *xhdn*
* Created *Fri* May 24 14:13:38 *CST* 2019 by *xhdn*