MODULE *StateAWSet*

EXTENDS *AWSet*, *FiniteSets*
CONSTANTS *Read*(_), *InitMsg*

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 correctness:
doset, *doset*[*r*]: the set of updates generated by replica $r \in Replica$
delset, *delset*[*r*]: the set of updates delivered by replica $r \in Replica$
uincoming *uincoming*[*r*]: incoming channel for broadcasting/delivering updates at $r \in Replica$

$nVars \triangleq \langle incoming, lmsg \rangle$
$cVars \triangleq \langle doset, delset, uincoming \rangle$
$vars \triangleq \langle aset, tset, seq, nVars, cVars \rangle$

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

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

$TypeOK \triangleq$
$\land\ aset \in [Replica \to \text{SUBSET } Element]$
$\land\ tset \in [Replica \to \text{SUBSET } Element]$
$\land\ IntTypeOK$
$\land\ Correctness!CTypeOK$

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

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

$Remove(d, r) \triangleq$ $r \in Replica$ removes $d \in Data$

$\land$ LET $E \triangleq \{ele \in aset[r] : ele.d = d\}$ E may be empty
  IN $\land\ aset' = [aset$ EXCEPT $![r] = @ \setminus E]$
     $\land\ tset' = [tset$ EXCEPT $![r] = @ \cup E]$
$\land\ IntDo(r)$
$\land\ Correctness!StateCDo(r)$
$\land$ UNCHANGED $\langle nVars \rangle$

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

$Send(r) \triangleq$ $r \in Replica$ sends a message
  $\land\ Network!BNBroadcast(r, [aid \mapsto [r \mapsto r, seq \mapsto seq[r]],$
                                $A \mapsto aset[r], T \mapsto tset[r]])$
  $\land\ IntSend(r)$
  $\land\ Correctness!StateCSend(r)$
  $\land$ UNCHANGED $\langle aset, tset \rangle$

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

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

$Fairness \triangleq \forall\, r \in Replica : \mathrm{WF}_{vars}(Send(r)) \land \mathrm{WF}_{vars}(Deliver(r))$

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

---

* Modification History
* Last modified Sat *Aug* 31 16:08:31 *CST* 2019 by *xhdn*
* Created *Fri* May 24 14:13:38 *CST* 2019 by *xhdn*