MODULE *StateAWSet*

EXTENDS *Naturals*, *Sequences*, *Interface*

CONSTANTS

Data the set of data

VARIABLES

aset, *aset*[*r*]: set of active *Instance*(*s*) maintained by $r \in Replica$

tset, *tset*[*r*]: set of tombstone *Instance*(*s*) maintained by $r \in Replica$

network variables

incoming, *incoming*[*r*]: incoming messages at replica $r \in Replica$

lmsg

$Nvars \triangleq \langle incoming, lmsg \rangle$

$vars \triangleq \langle aset, tset, seq, Nvars, SECvars \rangle$

$Element \triangleq [d : Data, r : Replica, k : Nat]$

$Msg \triangleq [r : Replica, seq : Nat, update : \text{SUBSET } Uid, A : \text{SUBSET } Element, T : \text{SUBSET } Element]$

Any *Network*

$Network \triangleq \text{INSTANCE } BasicNetwork$

$TypeOK \triangleq$

$\land \quad aset \in [Replica \to \text{SUBSET } Element]$

$\land \quad tset \in [Replica \to \text{SUBSET } Element]$

$\land \quad IntTypeOK$

$Init \triangleq$ initial state

$\land\, aset = [r \in Replica \mapsto \{\}]$

$\land\, tset = [r \in Replica \mapsto \{\}]$

$\land\, Network!NInit$

$\land\, IntInit$

$Send(r) \triangleq$ $r \in Replica$ send a message

$\land\, Network!NBroadcast(r,\ [r \mapsto r,\ seq \mapsto seq[r],\ update \mapsto StateUpdate(r),\ A \mapsto aset[r],\ T \mapsto tset[r]])$

$\land\, IntSend(r, [r \mapsto r,\ seq \mapsto seq[r],\ update \mapsto StateUpdate(r),\ A \mapsto aset[r],\ T \mapsto tset[r]])$

$\land\, \text{UNCHANGED } \langle aset, tset \rangle$

$Receive(r) \triangleq$ $r \in Replica$ receive a message

$\land \quad Network!NDeliver(r)$

$\land \quad IntReceive(r, lmsg')$

$\land \quad tset' = [tset \text{ EXCEPT } ![r] = @ \cup lmsg'.T]$

$\land \quad aset' = [aset \text{ EXCEPT } ![r] = (@ \cup lmsg'.A) \setminus tset'[r]]$

$\land \quad \text{UNCHANGED } \langle\rangle$

$Add(d,\, r) \triangleq$ $r \in Replica$ add $d \in Data$
$\quad \land aset' = [aset \text{ EXCEPT } ![r] = @ \cup \{[d \mapsto d,\, r \mapsto r,\, k \mapsto seq[r]]\}]$
$\quad \land IntDo(r)$
$\quad \land Network!NDo$
$\quad \land \text{UNCHANGED } \langle tset \rangle$

$Remove(d,\, r) \triangleq$ $r \in Replica$ remove $d \in Data$
$\quad \land \text{LET } E \triangleq \{ele \in aset[r] : ele.d = d\}$
$\quad\quad \text{IN} \quad \land aset' = [aset \text{ EXCEPT } ![r] = @ \setminus E]$
$\quad\quad\quad\quad \land tset' = [tset \text{ EXCEPT } ![r] = @ \cup E]$
$\quad \land IntDo(r)$
$\quad \land Network!NDo$

$Read(r) \triangleq$ read the state of $r \in Replica$
$\quad \{ins.d : ins \in aset[r]\}$

$Do(r) \triangleq$ operations
$\quad \exists\, a \in Data : Add(a,\, r) \lor Remove(a,\, r)$

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

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

SEC: Strong Eventual Consistency
$SEC \triangleq \forall\, r1,\, r2 \in Replica :$
$\quad\quad Sameupdate(r1,\, r2) \Rightarrow Read(r1) = Read(r2)$

* Modification History
* Last modified Sun *Jun* 09 22:57:52 *CST* 2019 by *xhdn*
* Created *Fri* May 24 14:13:38 *CST* 2019 by *xhdn*