MODULE *OpAWSet*

CONSTANTS

$Data$ the set of data

VARIABLES

$set,$ $sSet[r]$: set of active $Element(s)$ maintained by $r \in Replica$

$abuf,$ $abuf[r]$: buffer of $Element(s)$ added maintained by $r \in Replica$

$rbuf$ $rbuf[r]$: buffer of $Element(s)$ removed maintained by $r \in Replica$

$Element \triangleq [d : Data, r : Replica, k : Nat]$ the set of elements

$Network \triangleq$ INSTANCE $ReliableCausalNetwork$ instance a reliable causal network

$TypeOK \triangleq$

$\land \; set \in [Replica \rightarrow \text{SUBSET } Element]$

$\land \; abuf \in [Replica \rightarrow \text{SUBSET } Element]$

$\land \; rbuf \in [Replica \rightarrow \text{SUBSET } Element]$

$Init \triangleq \ldots$ initial state

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

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

$Add(d, r) \triangleq \ldots$ $r \in Replica$ add $d \in Data$

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

$Do(r) \triangleq$ operations

$\exists\, d \in Data : Add(d, r) \lor Remove(d, r)$

$Next \triangleq$ next-state relation

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

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

1
1
1
1
1
1
1
1
1
1
1
1

```
$Add(d, r) \triangleq$    $r \in Replica$ add $d \in Data$
        $\land\ set' \ = [set \text{ EXCEPT } ![r] = @ \cup \{[d \mapsto d,\ r \mapsto r,\ k \mapsto seq[r]]\}]$
        $\land\ abuf' = [abuf \text{ EXCEPT } ![r] = @ \cup \{[d \mapsto d,\ r \mapsto r,\ k \mapsto seq[r]]\}]$

$Remove(d, r) \triangleq$    $r \in Replica$ remove $d \in Data$
      $\land\ \{ele \in set[r] : ele.d = d\} \neq \{\}$
      $\land\ \text{LET } E \triangleq \{ele \in set[r] : ele.d = d\}$
        $\text{IN}\ \ \land\ set' = [set \text{ EXCEPT } ![r] = @ \setminus E]$
              $\land\ rbuf' = [rbuf \text{ EXCEPT } ![r] = @ \cup E]$

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

$Do(r) \triangleq$    operations
        $\exists\, d \in Data : Add(d, r) \lor Remove(d, r)$
--------------------------------------------------------------------------------
$Send(r) \triangleq$    $r \in Replica$ send a message
        $\land\ Network!RCBroadcast(r, [r \mapsto r,\ seq \mapsto seq[r],\ update \ \mapsto OpUpdate(r),$
                      $vc \mapsto [vc \text{ EXCEPT } ![r][r] = @ + 1][r],\ abuf \mapsto abuf[r],\ rbuf \mapsto rbuf[r]])$
        $\land\ abuf' = [abuf \text{ EXCEPT } ![r] = \{\}]$
        $\land\ rbuf' = [rbuf \text{ EXCEPT } ![r] = \{\}]$

$Receive(r) \triangleq$    $r \in Replica$ receive a message
      $\land\ \ Network!RCDeliver(r)$
      $\land\ \ set' = [set \text{ EXCEPT } ![r] = (@ \cup lmsg'.abuf) \setminus lmsg'.rbuf]$
================================================================================
```

* Modification History

* Last modified *Mon Jun* 10 02:23:45 *CST* 2019 by *xhdn*

* Created *Mon Jun* 10 00:18:41 *CST* 2019 by *xhdn*