

MODULE <i>RGA</i>	
EXTENDS <i>Integers, Sequences, Naturals, TLC, InsertTree, SEC</i>	
VARIABLES <i>tree</i> , <i>tomb</i> , <i>insbuf</i> , <i>tombbuf</i> , <i>chins</i> , <i>seq</i> , <i>incoming</i> , network variable <i>msg</i> , network variable <i>messageset</i> , network variable <i>vc</i> network variable	
<i>vars</i> \triangleq $\langle tree, tomb, insbuf, tombbuf, chins, seq, incoming, msg, messageset, vc, SECvars \rangle$	
<i>vector</i> \triangleq $[Replica \rightarrow Nat]$	
<i>Msg</i> \triangleq $[r : Replica, vc : vector, seq : Nat, update : SUBSET Update, tombbuf : SUBSET node, insbuf : SUBSET Char]$	
<i>List</i> $\triangleq Seq(Char)$	
RECURSIVE <i>maxtime</i> (-, -) <i>maxtime</i> (<i>tr</i> , <i>curmax</i>) \triangleq IF <i>tr</i> = {} THEN <i>curmax</i> ELSE LET <i>t</i> \triangleq CHOOSE <i>x</i> \in <i>tr</i> : TRUE IN <i>maxtime</i> (<i>tr</i> \ { <i>t</i> }, <i>Nummax</i> ((<i>t.ts</i>).time, <i>curmax</i>))	
<i>Network</i> \triangleq INSTANCE <i>CausalNetwork</i>	
<i>TypeOK</i> \triangleq \wedge <i>tree</i> $\in [Replica \rightarrow SUBSET node]$ \wedge <i>tomb</i> $\in [Replica \rightarrow SUBSET Char]$ \wedge <i>insbuf</i> $\in [Replica \rightarrow SUBSET node]$ \wedge <i>tombbuf</i> $\in [Replica \rightarrow SUBSET Char]$ \wedge <i>chins</i> $\in SUBSET Char$	
<i>Init</i> \triangleq \wedge <i>Network!</i> <i>Init</i> \wedge <i>SECInit</i> \wedge <i>tree</i> = $[r \in Replica \mapsto \{\}]$ \wedge <i>tomb</i> = $[r \in Replica \mapsto \{\}]$ \wedge <i>insbuf</i> = $[r \in Replica \mapsto \{\}]$ \wedge <i>tombbuf</i> = $[r \in Replica \mapsto \{\}]$	

$$\begin{aligned} &\wedge chins = Char \\ &\wedge seq = [r \in Replica \mapsto 0] \end{aligned}$$

$$\begin{aligned} DoIns(r) &\triangleq \\ &\exists ins \in node : \\ &\quad \wedge ins.parent \in Readtree2set(tree[r]) \cup \{\text{"o"}\} \\ &\quad \wedge ins.ts = [r \mapsto r, time \mapsto maxtime(tree[r], 1)] \\ &\quad \wedge ins.ch \in chins \\ &\quad \wedge chins' = chins \setminus \{ins.ch\} \\ &\quad \wedge tree' = [tree \text{ EXCEPT } ![r] = @ \cup \{ins\}] \\ &\quad \wedge insbuf' = [insbuf \text{ EXCEPT } ![r] = @ \cup \{ins\}] \\ &\quad \wedge seq' = [seq \text{ EXCEPT } ![r] = @ + 1] \\ &\quad \wedge SECUupdate(r, seq[r]) \\ &\quad \wedge \text{UNCHANGED } \langle incoming, msg, messageset, vc, tomb, tombbuf \rangle \end{aligned}$$

$$\begin{aligned} DoDel(r) &\triangleq \\ &\exists del \in Char : \\ &\quad \wedge del \in Readtree2set(tree[r]) \\ &\quad \wedge \neg del \in tomb[r] \\ &\quad \wedge tomb' = [tomb \text{ EXCEPT } ![r] = @ \cup \{del\}] \\ &\quad \wedge tombbuf' = [tombbuf \text{ EXCEPT } ![r] = @ \cup \{del\}] \\ &\quad \wedge seq' = [seq \text{ EXCEPT } ![r] = @ + 1] \\ &\quad \wedge SECUupdate(r, seq[r]) \\ &\quad \wedge \text{UNCHANGED } \langle chins, tree, insbuf, incoming, msg, messageset, vc \rangle \end{aligned}$$

do transitions

$$\begin{aligned} Do(r) &\triangleq \\ &\vee DoIns(r) \\ &\vee DoDel(r) \end{aligned}$$

send transitions

$$\begin{aligned} Send(r) &\triangleq \\ &\quad \wedge \vee tombbuf[r] \neq \{\} \\ &\quad \quad \vee insbuf[r] \neq \{\} \\ &\quad \wedge Network!Broadcast(r, [r \mapsto r, seq \mapsto seq[r], \\ &\quad \quad \quad vc \mapsto [vc \text{ EXCEPT } ![r][r] = @ + 1][r], update \mapsto OpUpdate(r), \\ &\quad \quad \quad tombbuf \mapsto tombbuf[r], insbuf \mapsto insbuf[r]]) \\ &\quad \wedge SECSend(r) \\ &\quad \wedge tombbuf' = [tombbuf \text{ EXCEPT } ![r] = \{\}] \\ &\quad \wedge insbuf' = [insbuf \text{ EXCEPT } ![r] = \{\}] \\ &\quad \wedge \text{UNCHANGED } \langle chins, seq, tree, tomb \rangle \end{aligned}$$

receive transitions

$$\begin{aligned} Receive(r) &\triangleq \\ &\quad \wedge Network!Deliver(r) \end{aligned}$$

$$\begin{array}{l}
\wedge \text{ SEC} \text{Deliver}(r, \text{msg}'[r]) \\
\wedge \text{ tree}' = [\text{tree} \text{ EXCEPT } ![r] = @ \cup \text{msg}'[r].\text{insbuf}] \\
\wedge \text{ tomb}' = [\text{tomb} \text{ EXCEPT } ![r] = @ \cup \text{msg}'[r].\text{tombbuf}] \\
\wedge \text{ UNCHANGED } \langle \text{chins}, \text{seq}, \text{tombbuf}, \text{insbuf} \rangle
\end{array}$$

$$\begin{array}{l}
\text{Next} \triangleq \\
\exists r \in \text{Replica} : \text{Do}(r) \vee \text{Send}(r) \vee \text{Receive}(r)
\end{array}$$

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

eventual consistency

$$\begin{array}{l}
\text{EmptyBuffer} \triangleq \text{tombbuf} = [r \in \text{Replica} \mapsto \{\}] \wedge \text{insbuf} = [r \in \text{Replica} \mapsto \{\}] \\
\text{EC} \triangleq \text{Network!EmptyChannel} \wedge \text{EmptyBuffer} \\
\Rightarrow \forall r1, r2 \in \text{Replica} : \\
\quad \text{Readtree2list}(\text{tree}[r1], \text{"o"}, \text{tomb}[r1], \{\}) = \text{Readtree2list}(\text{tree}[r2], \text{"o"}, \text{tomb}[r2], \{\}) \\
\text{SEC} \triangleq \exists r1, r2 \in \text{Replica} : \text{Sameupdate}(r1, r2) \\
\Rightarrow \text{Readtree2list}(\text{tree}[r1], \text{"o"}, \text{tomb}[r1], \{\}) = \text{Readtree2list}(\text{tree}[r2], \text{"o"}, \text{tomb}[r2], \{\})
\end{array}$$

\ * Modification History
\ * Last modified Tue May 07 01:27:02 CST 2019 by xhdn
\ * Last modified Mon May 06 16:52:19 CST 2019 by jywelling
\ * Last modified Thu Jan 10 15:34:04 CST 2019 by jywelling
\ * Created Tue Nov 06 15:55:23 CST 2018 by xhdn