

```

1  ┌────────────────────────── MODULE CassandraPaxos ───────────────────┐
2  EXTENDS Integers, Sequences, FiniteSets
4  Maximum(S)  $\triangleq$ 
    If S is a set of numbers, then this define Maximum(S) to be the maximum of those numbers,
    or  $-1$  if S is empty.
9  IF S = {} THEN  $-1$ 
10 ELSE CHOOSE  $n \in S : \forall m \in S : n \geq m$ 
12 Same(S)  $\triangleq$ 
    If S is not empty, then this define Same(S) to be the if or not the element of S is same. It is
    designed for determine whether the results of Read are the same
18  $\wedge S \neq \{\}$ 
19  $\wedge \exists n \in S : \forall m \in S : n = m$ 
21 CONSTANTS Value, Acceptor, Quorum, Operator
23 ASSUME  $\wedge \forall Q \in \text{Quorum} : Q \subseteq \text{Acceptor}$ 
24  $\wedge \forall Q1, Q2 \in \text{Quorum} : Q1 \cap Q2 \neq \{\}$ 
26 Ballot  $\triangleq \text{Nat}$ 
27 Version  $\triangleq \text{Nat}$ 
28 None  $\triangleq$  CHOOSE  $v : v \notin \text{Value}$ 
30 MeetCondition(ev, val)  $\triangleq$   $\vee val = \text{None}$ 
31  $\vee$  CASE Operator = ">"  $\rightarrow val > ev$ 
32  $\square$  Operator = "<"  $\rightarrow val < ev$ 
33  $\square$  Operator = "="  $\rightarrow val = ev$ 
34  $\square$  Operator = ">="  $\rightarrow val \geq ev$ 
35  $\square$  Operator = "<="  $\rightarrow val \leq ev$ 
36  $\square$  Operator = "/="  $\rightarrow val \neq ev$ 
37  $\square$  OTHER  $\rightarrow \text{FALSE}$ 
38 Message  $\triangleq$ 
39  $[type : \{\text{"Prepare"}\}, bal : \text{Ballot}]$ 
40  $\cup [type : \{\text{"Promise"}\}, acc : \text{Acceptor}, bal : \text{Ballot},$ 
41  $maxAccBal : \text{Ballot} \cup \{-1\}, maxAccVal : \text{Value} \cup \{\text{None}\},$ 
42  $maxComBal : \text{Ballot} \cup \{-1\}, maxComVal : \text{Value} \cup \{\text{None}\}]$ 
43  $\cup [type : \{\text{"Read"}\}, bal : \text{Ballot}]$ 
44  $\cup [type : \{\text{"Result"}\}, bal : \text{Ballot}, acc : \text{Acceptor},$ 
45  $value : \text{Value} \cup \{\text{None}\}, version : \text{Version}]$ 
46  $\cup [type : \{\text{"Repair"}\}, value : \text{Value} \cup \{\text{None}\}, version : \text{Version}]$ 
47  $\cup [type : \{\text{"Propose"}\}, bal : \text{Ballot}, val : \text{Value}]$ 
48  $\cup [type : \{\text{"Accept"}\}, acc : \text{Acceptor}, bal : \text{Ballot}, val : \text{Value}]$ 
49  $\cup [type : \{\text{"Commit"}\}, bal : \text{Ballot}, val : \text{Value}]$ 
50  $\cup [type : \{\text{"Ack"}\}, acc : \text{Acceptor}, bal : \text{Ballot}, val : \text{Value}]$ 
51  $\cup [type : \{\text{"Terminate"}\}, bal : \text{Ballot}]$ 

```

```

53 VARIABLES  $maxBal, maxAccBal, maxAccVal, maxComBal,$ 
54            $maxComVal, msgs, dataResult, balValue$ 
55  $vars \triangleq \langle maxBal, maxAccBal, maxAccVal, maxComBal,$ 
56            $maxComVal, msgs, dataResult, balValue \rangle$ 

58  $TypeOK \triangleq \wedge maxBal \in [Acceptor \rightarrow Ballot \cup \{-1\}]$ 
59            $\wedge maxAccBal \in [Acceptor \rightarrow Ballot \cup \{-1\}]$ 
60            $\wedge maxAccVal \in [Acceptor \rightarrow Value \cup \{None\}]$ 
61            $\wedge maxComBal \in [Acceptor \rightarrow Ballot \cup \{-1\}]$ 
62            $\wedge maxComVal \in [Acceptor \rightarrow Value \cup \{None\}]$ 
63            $\wedge dataResult \in [Acceptor \rightarrow [value : Value \cup \{None\},$ 
64            $version : Version]]$ 
65            $\wedge msgs \subseteq Message$ 
66            $\wedge balValue \in [Ballot \rightarrow [expVal : Value \cup \{None\},$ 
67            $setVal : Value \cup \{None\}]]$ 

69  $Init \triangleq \wedge maxBal = [a \in Acceptor \mapsto -1]$ 
70            $\wedge maxAccBal = [a \in Acceptor \mapsto -1]$ 
71            $\wedge maxAccVal = [a \in Acceptor \mapsto None]$ 
72            $\wedge maxComBal = [a \in Acceptor \mapsto -1]$ 
73            $\wedge maxComVal = [a \in Acceptor \mapsto None]$ 
74            $\wedge dataResult = [a \in Acceptor \mapsto [value \mapsto None, version \mapsto 0]]$ 
75            $\wedge msgs = \{\}$ 
76            $\wedge balValue = [b \in Ballot \mapsto [expVal \mapsto None, setVal \mapsto None]]$ 

79  $Send(m) \triangleq msgs' = msgs \cup \{m\}$ 

81  $CAS(ev, sv, b) \triangleq \wedge \neg \exists m \in msgs : m.type = \text{"Prepare"} \wedge m.bal = b$ 
82            $\wedge Send([type \mapsto \text{"Prepare"}, bal \mapsto b])$ 
83            $\wedge balValue' = [balValue \text{ EXCEPT } ![b] =$ 
84            $[expVal \mapsto ev, setVal \mapsto sv]]$ 
85            $\wedge \text{UNCHANGED } \langle maxBal, maxAccBal, maxAccVal, maxComBal,$ 
86            $maxComVal, dataResult \rangle$ 

88  $Promise(a) \triangleq$ 
89    $\wedge \exists m \in msgs :$ 
90      $\wedge m.type = \text{"Prepare"}$ 
91      $\wedge m.bal > maxBal[a]$ 
92      $\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$ 
93      $\wedge Send([type \mapsto \text{"Promise"}, acc \mapsto a, bal \mapsto m.bal,$ 
94            $maxAccBal \mapsto maxAccBal[a], maxAccVal \mapsto maxAccVal[a],$ 
95            $maxComBal \mapsto maxComBal[a], maxComVal \mapsto maxComVal[a]$ 
96            $])$ 

98    $\wedge \text{UNCHANGED } \langle maxAccBal, maxAccVal, maxComBal, maxComVal,$ 
99    $dataResult, balValue \rangle$ 

```

```

102  $Propose(b) \triangleq \wedge \neg \exists m \in msgs : m.type = \text{"Propose"} \wedge m.bal = b$ 
103  $\wedge \exists Q \in Quorum :$ 
104  $LET \ Qmset \triangleq \{m \in msgs : \wedge m.type = \text{"Promise"}$ 
105  $\wedge m.acc \in Q$ 
106  $\wedge m.bal = b\}$ 
107  $maxAccbal \triangleq Maximum(\{m.maxAccBal : m \in Qmset\})$ 
108  $maxCombal \triangleq Maximum(\{m.maxComBal : m \in Qmset\})$ 
109  $preValue \triangleq (CHOOSE \ m \in Qmset : m.maxAccBal = maxAccbal).maxAccVal$ 
110  $IN \ \wedge \forall a \in Q : \exists m \in Qmset : m.acc = a$ 
111  $\wedge IF \ maxAccbal > maxCombal \ THEN \ Send([type \mapsto \text{"Propose"},$ 
112  $bal \mapsto b, val \mapsto preValue])$ 
113  $ELSE \ Send([type \mapsto \text{"Read"}, bal \mapsto b])$ 
114  $\wedge UNCHANGED \langle maxBal, maxAccBal, maxAccVal, maxComBal, maxComVal,$ 
115  $dataResult, balValue \rangle$ 

118  $Read(a) \triangleq \wedge \exists m \in msgs : \wedge m.type = \text{"Read"}$ 
119  $\wedge Send([type \mapsto \text{"Result"}, acc \mapsto a, bal \mapsto m.bal,$ 
120  $value \mapsto dataResult[a].value,$ 
121  $version \mapsto dataResult[a].version])$ 
122  $\wedge UNCHANGED \langle maxBal, maxAccBal, maxAccVal, maxComBal, maxComVal,$ 
123  $dataResult, balValue \rangle$ 

125  $Result(b) \triangleq \wedge \exists Q \in Quorum :$ 
126  $LET \ QRmset \triangleq \{m \in msgs : \wedge m.type = \text{"Result"}$ 
127  $\wedge m.acc \in Q$ 
128  $\wedge m.bal = b\}$ 
129  $QResult \triangleq \{m.value : m \in QRmset\}$ 
130  $maxVersion \triangleq Maximum(\{m.version : m \in QRmset\})$ 
131  $maxValue \triangleq (CHOOSE \ m \in QRmset : m.version = maxVersion).value$ 
132  $IN \ \wedge \forall a \in Q : \exists m \in QRmset : m.acc = a$ 
133  $\wedge IF \ MeetCondition(balValue[b].expVal, maxVersion)$ 
134  $THEN \ Send([type \mapsto \text{"Propose"}, bal \mapsto b,$ 
135  $val \mapsto balValue[b].setVal])$ 
136  $ELSE \ Send([type \mapsto \text{"Terminate"}, bal \mapsto b])$ 

138  $\wedge IF \ \neg Same(QResult) \ THEN \ Send([type \mapsto \text{"Repair"},$ 
139  $value \mapsto maxVersion,$ 
140  $version \mapsto maxVersion])$ 
141  $\wedge UNCHANGED \langle maxBal, maxAccBal, maxAccVal, maxComBal, maxComVal,$ 
142  $dataResult, balValue \rangle$ 

```

```

*****
Repair(a)  $\triangleq \wedge \exists m \in msgs : \wedge m.type = \text{"Repair"}$ 
 $\wedge dataResult' = [dataResult \ EXCEPT \ ![a] = [value \mapsto m.value, version \mapsto$ 
 $m.version]]$ 
 $\wedge UNCHANGED \langle maxBal, maxAccBal, maxAccVal, maxComBal, maxComVal,$ 

```

```

154  $Accept(a) \triangleq \wedge \exists m \in msgs : \wedge m.type = \text{"Propose"}$ 
155  $\wedge maxBal[a] = m.bal$ 
156  $\wedge maxBal[a] \leq m.bal$ 
157  $\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$ 
158  $\wedge maxAccBal' = [maxAccBal \text{ EXCEPT } ![a] = m.bal]$ 
159  $\wedge maxAccVal' = [maxAccVal \text{ EXCEPT } ![a] = m.val]$ 
160  $\wedge Send([type \mapsto \text{"Accept"}, bal \mapsto m.bal,$ 
161  $val \mapsto m.val, acc \mapsto a])$ 
162  $\wedge \text{UNCHANGED } \langle maxComBal, maxComVal, dataResult, balValue \rangle$ 
163  $\wedge \text{UNCHANGED } \langle maxBal, maxComBal, maxComVal, dataResult, balValue \rangle$ 

166  $Commit(b) \triangleq \wedge \neg \exists m \in msgs : m.type = \text{"Commit"} \wedge m.bal = b$ 
167  $\wedge \exists Q \in Quorum :$ 
168  $\text{LET } QAmset \triangleq \{m \in msgs : \wedge m.type = \text{"Accept"}$ 
169  $\wedge m.acc \in Q$ 
170  $\wedge m.bal = b\}$ 
171  $\text{IN } \wedge \forall a \in Q : \exists m \in QAmset : m.acc = a$ 
172  $\wedge Send([type \mapsto \text{"Commit"}, bal \mapsto b,$ 
173  $val \mapsto balValue[b].setVal])$ 
174  $\wedge \text{UNCHANGED } \langle maxBal, maxAccBal, maxAccVal, maxComBal,$ 
175  $maxComVal, dataResult, balValue \rangle$ 

177  $Ack(a) \triangleq \wedge \exists m \in msgs : \wedge m.type = \text{"Commit"}$ 
178  $\wedge maxBal[a] \leq m.bal$ 
179  $\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$ 
180  $\wedge maxBal[a] = m.bal$ 
181  $\wedge maxComBal' = [maxComBal \text{ EXCEPT } ![a] = m.bal]$ 
182  $\wedge maxComVal' = [maxComVal \text{ EXCEPT } ![a] = m.val]$ 
183  $\wedge dataResult' = [dataResult \text{ EXCEPT } ![a] =$ 
184  $[value \mapsto m.val, version \mapsto (@.version + 1)]]$ 
185  $\wedge Send([type \mapsto \text{"Ack"}, bal \mapsto m.bal,$ 
186  $val \mapsto m.val, acc \mapsto a])$ 
187  $\wedge \text{UNCHANGED } \langle maxAccBal, maxAccVal, balValue \rangle$ 
188  $\wedge \text{UNCHANGED } \langle maxAccBal, maxAccVal, dataResult, balValue \rangle$ 
189  $\wedge \text{UNCHANGED } \langle maxBal, maxAccBal, maxAccVal, dataResult, balValue \rangle$ 

191  $Next \triangleq \vee \exists ev, sv \in Value, b \in Ballot : CAS(ev, sv, b)$ 
192  $\vee \exists b \in Ballot : \vee Propose(b)$ 
193  $\vee Result(b)$ 
194  $\vee Commit(b)$ 
195  $\vee \exists a \in Acceptor : \vee Promise(a)$ 
196  $\vee Accept(a)$ 
197  $\vee Ack(a)$ 

```

```

198                                      $\vee Read(a)$ 
199                                      $\vee Repair(a)$ 
201   $Spec \triangleq Init \wedge \Box[Next]_{vars}$ 
203  ┌───────────────────────────────────────────────────────────────────────────────────┐
    \| * Modification History
    \| * Last modified Thu Mar 03 17:37:19 CST 2022 by LENOVO
    \| * Created Thu Dec 08 10:19:29 CST 2021 by LENOVO

```