

```

1  |----- MODULE CJupiter -----|
   | Model of our own CJupiter protocol. |
5  | EXTENDS StateSpace, JupiterSerial |
6  |-----|
7  | VARIABLES
8  |     css      |css[r]: the n-ary ordered state space at replica r ∈ Replica|
10 | vars ≜ ⟨intVars, ctxVars, serialVars, css⟩
11 |-----|
12 | TypeOK ≜
13 |     ∧ TypeOKInt
14 |     ∧ TypeOKCtx
15 |     ∧ TypeOKSerial
16 |     ∧ Comm(Cop)!TypeOK
17 |     ∧ ∀ r ∈ Replica : IsSS(css[r])
18 |-----|
19 | Init ≜
20 |     ∧ InitInt
21 |     ∧ InitCtx
22 |     ∧ InitSerial
23 |     ∧ Comm(Cop)!Init
24 |     ∧ css = [r ∈ Replica ↦ EmptySS]
25 |-----|
   | xForm: Iteratively transform cop with a path through the css at replica r ∈ Replica, following |
   | the first edges. |
30 | xForm(cop, r) ≜
31 |     LET rcss ≜ css[r]
32 |     u ≜ Locate(cop, rcss)
33 |     v ≜ u ∪ {cop.oid}
34 |     RECURSIVE xFormHelper(−, −, −, −)
35 |     xFormHelper(uh, vh, coph, xcsc) ≜ |xcsc: eXtra css created during transformation|
36 |     IF uh = ds[r] THEN [xcsc ↦ xcsc, xcop ↦ coph]
37 |     ELSE LET fedge ≜ CHOOSE e ∈ rcsc.edge :
38 |         ∧ e.from = uh
39 |         ∧ ∀ uhe ∈ rcsc.edge \ {e} :
40 |             (uhe.from = uh) ⇒ tb(e.cop.oid, uhe.cop.oid, serial[r])
41 |         uprime ≜ fedge.to
42 |         fcop ≜ fedge.cop
43 |         coph2fcop ≜ COT(coph, fcop)
44 |         fcop2coph ≜ COT(fcop, coph)
45 |         vprime ≜ vh ∪ {fcop.oid}
46 |     IN xFormHelper(uprime, vprime, coph2fcop,
47 |         xcsc ⊕ [node ↦ {vprime},
48 |             edge ↦ {[from ↦ vh, to ↦ vprime, cop ↦ fcop2coph],
49 |                 [from ↦ uprime, to ↦ vprime, cop ↦ coph2fcop]}])

```

```

50      IN    $xFormHelper(u, v, cop, [node \mapsto \{v\}, edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto cop]\}])$ 
      Perform cop at replica  $r \in Replica$ .
54   $Perform(cop, r) \triangleq$ 
55      LET  $xform \triangleq xForm(cop, r)$   $xform: [xcss, xcop]$ 
56      IN    $\wedge css' = [css \text{ EXCEPT } ![r] = @ \oplus xform.xcss]$ 
57           $\wedge state' = [state \text{ EXCEPT } ![r] = Apply(xform.xcop.op, @)]$ 
58  |-----|
      Client  $c \in Client$  issues an operation  $op$ .
62   $DoOp(c, op) \triangleq$ 
63       $\wedge$  LET  $cop \triangleq [op \mapsto op, oid \mapsto [c \mapsto c, seq \mapsto cseq'[c]], ctx \mapsto ds[c]]$ 
64      IN    $\wedge Perform(cop, c)$ 
65           $\wedge Comm(Cop)!CSend(cop)$ 

67   $DoIns(c) \triangleq$ 
68       $\exists ins \in \{op \in Ins : op.pos \in 1 \dots (Len(state[c]) + 1) \wedge op.ch \in chins \wedge op.pr = Priority[c]\} :$ 
69           $\wedge DoOp(c, ins)$ 
70           $\wedge chins' = chins \setminus \{ins.ch\}$ 

72   $DoDel(c) \triangleq$ 
73       $\exists del \in \{op \in Del : op.pos \in 1 \dots Len(state[c])\} :$ 
74           $\wedge DoOp(c, del)$ 
75           $\wedge \text{UNCHANGED } chins$ 

77   $Do(c) \triangleq$ 
78       $\wedge DoCtx(c)$ 
79       $\wedge DoSerial(c)$ 
80       $\wedge \vee DoIns(c)$ 
81       $\vee DoDel(c)$ 
      Client  $c \in Client$  receives a message from the Server.
85   $Rev(c) \triangleq$ 
86       $\wedge Comm(Cop)!CRev(c)$ 
87       $\wedge Perform(Head(cincoming[c]), c)$ 
88       $\wedge RevSerial(c)$ 
89       $\wedge RevCtx(c)$ 
90       $\wedge \text{UNCHANGED } chins$ 
91  |-----|
      The Server receives a message.
95   $SRev \triangleq$ 
96       $\wedge Comm(Cop)!SRev$ 
97       $\wedge$  LET  $cop \triangleq Head(sincoming)$ 
98      IN    $\wedge Perform(cop, Server)$ 
99           $\wedge Comm(Cop)!SSendSame(cop.oid.c, cop)$   $\text{broadcast the original operation}$ 
100       $\wedge SRevSerial$ 
101       $\wedge SRevCtx$ 
102       $\wedge \text{UNCHANGED } chins$ 

```

```

103 |-----|
104 Next  $\triangleq$ 
105      $\vee \exists c \in \textit{Client} : \textit{Do}(c) \vee \textit{Rev}(c)$ 
106      $\vee \textit{SRev}$ 
108 Fairness  $\triangleq$  There is no requirement that the clients ever generate operations.
109      $\text{WF}_{\textit{vars}}(\textit{SRev} \vee \exists c \in \textit{Client} : \textit{Rev}(c))$ 
111 Spec  $\triangleq \textit{Init} \wedge \Box[\textit{Next}]_{\textit{vars}} \wedge \textit{Fairness}$  (We care more about safety.)
112 |-----|
113 Compactness  $\triangleq$  The compactness of CJupiter: the CSSes at all replicas are the same.
114      $\textit{Comm}(\textit{Cop})! \textit{EmptyChannel} \Rightarrow \textit{Cardinality}(\textit{Range}(\textit{css})) = 1$ 
116 THEOREM  $\textit{Spec} \Rightarrow \textit{Compactness}$ 
117 |-----|
    \ * Modification History
    \ * Last modified Mon Dec 31 11:02:07 CST 2018 by hengxin
    \ * Created Sat Sep 01 11:08:00 CST 2018 by hengxin

```