

```

1 |----- MODULE CJupiter -----|
  Specification of CJupiter; see Wei@OPODIS'18.
5 | EXTENDS StateSpace, JupiterSerial |
6 |-----|
7 | VARIABLES
8   css      css[r]: the n-ary ordered state space at replica  $r \in Replica$ 
10  vars  $\triangleq \langle intVars, ctxVars, serialVars, css \rangle$ 
11 |-----|
12  TypeOK  $\triangleq$ 
13     $\wedge$  TypeOKInt
14     $\wedge$  TypeOKCtx
15     $\wedge$  TypeOKSerial
16     $\wedge$  Comm(Cop)!TypeOK
17     $\wedge$   $\forall r \in Replica : IsSS(css[r])$ 
18 |-----|
19  Init  $\triangleq$ 
20     $\wedge$  InitInt
21     $\wedge$  InitCtx
22     $\wedge$  InitSerial
23     $\wedge$  Comm(Cop)!Init
24     $\wedge$   $css = [r \in Replica \mapsto EmptySS]$ 
25 |-----|
    xForm: Iteratively transform cop with a path through the css at replica  $r \in Replica$ , following
    the first edges.
30  xForm(cop, r)  $\triangleq$ 
31    LET rcss  $\triangleq$  css[r]
32    u  $\triangleq$  Locate(cop, rcss)
33    v  $\triangleq$  u  $\cup$  {cop.oid}
34    RECURSIVE xFormHelper(–, –, –, –)
35    'h' stands for "helper"; xcsc: eXtra css created during transformation
36    xFormHelper(uh, vh, coph, xcsc)  $\triangleq$ 
37    IF uh = ds[r]
38      THEN [xcsc  $\mapsto$  xcsc, xcop  $\mapsto$  coph]
39      ELSE LET fedge  $\triangleq$  CHOOSE  $e \in rcsc.edge :$ 
40         $\wedge e.from = uh$ 
41         $\wedge \forall uhe \in rcsc.edge :$ 
42           $(uhe.from = uh \wedge uhe \neq e) \Rightarrow tb(e.cop.oid, uhe.cop.oid, serial[r])$ 
43        uprime  $\triangleq$  fedge.to
44        fcop  $\triangleq$  fedge.cop
45        coph2fcop  $\triangleq$  COT(coph, fcop)
46        fcop2coph  $\triangleq$  COT(fcop, coph)
47        vprime  $\triangleq$  vh  $\cup$  {fcop.oid}
48    IN  xFormHelper(uprime, vprime, coph2fcop,
49          xcsc  $\oplus$  [node  $\mapsto$  {vprime}],

```

```

50                                     edge  $\mapsto \{[from \mapsto vh, to \mapsto vprime, cop \mapsto fcop2coph],$ 
51                                      $[from \mapsto uprime, to \mapsto vprime, cop \mapsto coph2fcop]\}$ 
52     IN     $xFormHelper(u, v, cop, [node \mapsto \{v\}, edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto cop]\}])$ 
    Perform cop at replica  $r \in Replica$ .
56  $Perform(cop, r) \triangleq$ 
57     LET  $xform \triangleq xForm(cop, r)$   $xform: [xcss, xcop]$ 
58     IN     $\wedge css' = [css \text{ EXCEPT } ![r] = @ \oplus xform.xcss]$ 
59           $\wedge state' = [state \text{ EXCEPT } ![r] = Apply(xform.xcop.op, @)]$ 
60 |-----|
    Client  $c \in Client$  issues an operation  $op$ .
64  $DoOp(c, op) \triangleq$   $op$ : the raw operation generated by the client  $c \in Client$ 
65      $\wedge$  LET  $cop \triangleq [op \mapsto op, oid \mapsto [c \mapsto c, seq \mapsto cseq'[c]], ctx \mapsto ds[c]]$ 
66     IN     $\wedge Perform(cop, c)$ 
67      $\wedge Comm(Cop)!CSend(cop)$ 

69  $DoIns(c) \triangleq$ 
70      $\exists ins \in \{op \in Ins : op.pos \in 1 \dots (Len(state[c]) + 1) \wedge op.ch \in chins \wedge op.pr = Priority[c]\} :$ 
71      $\wedge DoOp(c, ins)$ 
72      $\wedge chins' = chins \setminus \{ins.ch\}$  We assume that all inserted elements are unique.

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

79  $Do(c) \triangleq$ 
80      $\wedge DoCtx(c)$ 
81      $\wedge DoSerial(c)$ 
82      $\wedge \vee DoIns(c)$ 
83      $\vee DoDel(c)$ 
    Client  $c \in Client$  receives a message from the Server.
87  $Rev(c) \triangleq$ 
88      $\wedge Comm(Cop)!CRev(c)$ 
89      $\wedge Perform(Head(cincoming[c]), c)$ 
90      $\wedge RevSerial(c)$ 
91      $\wedge RevCtx(c)$ 
92      $\wedge \text{UNCHANGED } chins$ 
93 |-----|
    The Server receives a message.
97  $SRev \triangleq$ 
98      $\wedge Comm(Cop)!SRev$ 
99      $\wedge$  LET  $cop \triangleq Head(sincoming)$ 
100    IN     $\wedge Perform(cop, Server)$ 
101           $\wedge Comm(Cop)!SSendSame(cop.oid.c, cop)$  broadcast the original operation
102     $\wedge SRevSerial$ 

```

```

103       $\wedge SRevCtx$ 
104       $\wedge \text{UNCHANGED } chins$ 
105  |-----|
106   $Next \triangleq$ 
107       $\vee \exists c \in Client : Do(c) \vee Rev(c)$ 
108       $\vee SRev$ 
109  Fairness: There is no requirement that the clients ever generate operations.
112   $Fairness \triangleq$ 
113       $WF_{vars}(SRev \vee \exists c \in Client : Rev(c))$ 
115   $Spec \triangleq Init \wedge \Box[Next]_{vars} \wedge Fairness$  (We care more about safety.)
116  |-----|
117  The compactness of CJupiter: the CSSes at all replicas are the same.
120   $Compactness \triangleq$ 
121       $Comm(Cop)!EmptyChannel \Rightarrow Cardinality(Range(css)) = 1$ 
123  THEOREM  $Spec \Rightarrow Compactness$ 
124  |-----|
    \ * Modification History
    \ * Last modified Tue Feb 05 10:55:47 CST 2019 by hengxin
    \ * Created Sat Sep 01 11:08:00 CST 2018 by hengxin

```