

```

1  ┌────────────────────────── MODULE CJupiter ───────────────────────────┐
    Specification of our own CJupiter protocol; see Wei@OPODIS'2018.
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     ∧ ∀ r ∈ Replica : IsSS(css[r])
17 ┌──────────────────────────────────────────────────────────────────────────┐
18 Init ≜
19     ∧ InitInt
20     ∧ InitCtx
21     ∧ InitSerial
22     ∧ css = [r ∈ Replica ↦ EmptySS]
23 ┌──────────────────────────────────────────────────────────────────────────┐
    Iteratively transform cop with a path in the css at replica r ∈ Replica, following the first edges.
28 xForm(r, cop) ≜
29     LET rcss ≜ css[r]
30     u ≜ Locate(cop, rcss)
31     v ≜ u ∪ {cop.oid}
32     RECURSIVE xFormHelper(u, u, u, u)
33     xFormHelper(uh, vh, coph, xcss) ≜ xcss: eXtra css created during transformation
34     IF uh = ds[r] THEN [xcop ↦ coph, xcss ↦ xcss]
35     ELSE LET fedge ≜ the first outgoing edge from uh
36         CHOOSE e ∈ rcss.edge :
37             ∧ e.from = uh
38             ∧ ∀ uhe ∈ rcss.edge \ {e} :
39                 (uhe.from = uh) ⇒ tb(e.cop.oid, uhe.cop.oid, serial[r])
40             uprime ≜ fedge.to
41             fcop ≜ fedge.cop
42             coph2fcop ≜ COT(coph, fcop)
43             fcop2coph ≜ COT(fcop, coph)
44             vprime ≜ vh ∪ {fcop.oid}
45         IN xFormHelper(uprime, vprime, coph2fcop,
46             xcss ⊕ [node ↦ {vprime},
47                 edge ↦ {[from ↦ vh, to ↦ vprime, cop ↦ fcop2coph],
48                     [from ↦ uprime, to ↦ vprime, cop ↦ coph2fcop]}])
49     IN xFormHelper(u, v, cop, [node ↦ {v}, edge ↦ {[from ↦ u, to ↦ v, cop ↦ cop]}])

```

```

51  $Perform(r, cop) \triangleq$ 
52   LET  $xform \triangleq xForm(r, cop)$   $xform: [xcop, xcss]$ 
53   IN    $\wedge css' = [css \text{ EXCEPT } ![r] = @ \oplus xform.xcss]$ 
54        $\wedge SetNewAop(r, xform.xcop.op)$ 
56  $ClientPerform(c, cop) \triangleq Perform(c, cop)$ 
58  $ServerPerform(cop) \triangleq$ 
59    $\wedge Perform(Server, cop)$ 
60    $\wedge Comm!SSendSame(ClientOf(cop), cop)$   $\text{broadcast the original operation}$ 
61 |-----|
62  $DoOp(c, op) \triangleq$ 
63   LET  $cop \triangleq [op \mapsto op, oid \mapsto [c \mapsto c, seq \mapsto cseq[c]], ctx \mapsto ds[c]]$ 
64   IN    $\wedge ClientPerform(c, cop)$ 
65        $\wedge Comm!CSend(cop)$ 
67  $Do(c) \triangleq$ 
68    $\wedge DoInt(DoOp, c)$ 
69    $\wedge DoCtx(c)$ 
70    $\wedge DoSerial(c)$ 
72  $Rev(c) \triangleq$ 
73    $\wedge RevInt(ClientPerform, c)$ 
74    $\wedge RevCtx(c)$ 
75    $\wedge RevSerial(c)$ 
77  $SRev \triangleq$ 
78    $\wedge SRevInt(ServerPerform)$ 
79    $\wedge SRevCtx$ 
80    $\wedge SRevSerial$ 
81 |-----|
82  $Next \triangleq$ 
83    $\vee \exists c \in Client : Do(c) \vee Rev(c)$ 
84    $\vee SRev$ 
86  $Fairness \triangleq$ 
87    $WF_{vars}(SRev \vee \exists c \in Client : Rev(c))$ 
89  $Spec \triangleq Init \wedge \Box[Next]_{vars} \wedge Fairness$ 
90 |-----|
91  $Compactness \triangleq$   $\text{Compactness of CJupiter: the CSSes at all replicas are the same.}$ 
92    $Comm!EmptyChannel \Rightarrow Cardinality(Range(css)) = 1$ 
94 THEOREM  $Spec \Rightarrow Compactness$ 
95 |-----|
  
```

\ * Modification History
 \ * Last modified Mon Jan 07 13:19:43 CST 2019 by hengxin
 \ * Created Sat Sep 01 11:08:00 CST 2018 by hengxin