

```

1  |----- MODULE AbsJupiter -----|
   | Abstract Jupiter, inspired by the COT algorithm proposed by Sun and Sun; see TPDS'2009. |
5  | EXTENDS JupiterSerial |
6  |-----|
7  VARIABLES
8    copss    copss[r]: the state space (i.e., a set) of Cops maintained at replica r ∈ Replica
10   vars ≜ ⟨intVars, ctxVars, serialVars, copss⟩
11 |-----|
12   TypeOK ≜
13     ∧ TypeOKInt
14     ∧ TypeOKCtx
15     ∧ TypeOKSerial
16     ∧ Comm(Cop)! TypeOK
17     ∧ copss ∈ [Replica → SUBSET Cop]
18 |-----|
19   Init ≜
20     ∧ InitInt
21     ∧ InitCtx
22     ∧ InitSerial
23     ∧ Comm(Cop)! Init
24     ∧ copss = [r ∈ Replica ↦ {}]
25 |-----|
26   RECURSIVE xForm(-, -)
27   xForm(cop, r) ≜
28     LET ctxDiff ≜ ds[r] \ cop.ctx THEOREM : cop.ctx ⊆ ds[r]
29     RECURSIVE xFormHelper(-, -, -)
30       xFormHelper(coph, ctxDiffh, copssr) ≜ copssr: state space generated during transformation
31       IF ctxDiffh = {} THEN [xcop ↦ coph, xcopss ↦ copssr]
32       ELSE LET foph ≜ CHOOSE op ∈ ctxDiffh : the first op in serial
33         ∃ opprime ∈ ctxDiffh \ {op} : tb(op, opprime, serial[r])
34         fcophDict ≜ {op ∈ copssr : op.oid = foph ∧ op.ctx = coph.ctx}
35         fcoph ≜ CHOOSE op ∈ fcophDict : TRUE THEOREM : Cardinality(fcophDict) = 1
36         xcoph ≜ COT(coph, fcoph)
37         xfcoph ≜ COT(fcoph, coph)
38         IN xFormHelper(xcoph, ctxDiffh \ {foph}, copssr ∪ {xcoph, xfcoph})
39     IN xFormHelper(cop, ctxDiff, copss[r])
41   Perform(cop, r) ≜
42     LET xform ≜ xForm(cop, r) [xcop, xcopss]
43     IN   ∧ state' = [state EXCEPT ![r] = Apply(xform.xcop.op, @)]
44         ∧ copss' = [copss EXCEPT ![r] = xform.xcopss ∪ {cop}]
45 |-----|
46   DoOp(c, op) ≜ Client c ∈ Client processes a locally generated operation op.
47     LET cop ≜ [op ↦ op, oid ↦ [c ↦ c, seq ↦ cseq'[c], ctx ↦ ds[c]]

```

```

48      IN   $\wedge \text{Perform}(\text{cop}, c)$ 
49       $\wedge \text{Comm}(\text{Cop})! \text{CSend}(\text{cop})$ 

51   $\text{Do}(c) \triangleq$ 
52       $\wedge \text{DoCtx}(c)$ 
53       $\wedge \text{DoSerial}(c)$ 
54       $\wedge \text{DoInt}(\text{DoOp}, c)$ 

56   $\text{Rev}(c) \triangleq$ 
57       $\wedge \text{Comm}(\text{Cop})! \text{CRev}(c)$ 
58       $\wedge \text{Perform}(\text{Head}(\text{cincoming}[c]), c)$ 
59       $\wedge \text{RevSerial}(c)$ 
60       $\wedge \text{RevCtx}(c)$ 
61       $\wedge \text{RevInt}(c)$ 

63   $\text{SRev} \triangleq$ 
64       $\wedge \text{Comm}(\text{Cop})! \text{SRev}$ 
65       $\wedge \text{LET } \text{cop} \triangleq \text{Head}(\text{sincoming})$ 
66      IN   $\wedge \text{Perform}(\text{cop}, \text{Server})$ 
67       $\wedge \text{Comm}(\text{Cop})! \text{SSendSame}(\text{cop.oid.c}, \text{cop})$ 
68       $\wedge \text{SRevSerial}$ 
69       $\wedge \text{SRevCtx}$ 
70       $\wedge \text{SRevInt}$ 

71  ───────────────────────────────────────────────────────────────────────────────────┐
72   $\text{Next} \triangleq$ 
73       $\vee \exists c \in \text{Client} : \text{Do}(c) \vee \text{Rev}(c)$ 
74       $\vee \text{SRev}$ 

76   $\text{Fairness} \triangleq$ 
77       $\text{WF}_{\text{vars}}(\text{SRev} \vee \exists c \in \text{Client} : \text{Rev}(c))$ 

79   $\text{Spec} \triangleq \text{Init} \wedge \Box[\text{Next}]_{\text{vars}} \wedge \text{Fairness}$ 
80  ───────────────────────────────────────────────────────────────────────────────────┐
81   $\text{Compactness} \triangleq$ 
82       $\text{Comm}(\text{Cop})! \text{EmptyChannel} \Rightarrow \text{Cardinality}(\text{Range}(\text{copss})) = 1$ 

84  THEOREM  $\text{Spec} \Rightarrow \text{Compactness}$ 
85  ───────────────────────────────────────────────────────────────────────────────────┐
    \ * Modification History
    \ * Last modified Mon Dec 31 20:27:49 CST 2018 by hengxin
    \ * Created Wed Dec 05 19:55:52 CST 2018 by hengxin

```