

```

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   ∧ copss' = [copss EXCEPT ! [r] = xform.xcopss ∪ {cop}]
44         ∧ SetNewAop(r, xform.xcop.op)
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]]

```

