```
- Module AbsJupiter -
 1 [
    Abstract Jupiter, inspired by the COT algorithm proposed by Sun and Sun; see TPDS'2009.
 5 EXTENDS JupiterSerial
 6 |
    VARIABLES
                   copss[r]: the state space (i.e., a set) of Cops maintained at replia r \in Replica
    vars \triangleq \langle intVars, ctxVars, serialVars, copss \rangle
10
    TypeOK \triangleq
12
               TypeOKInt
13
               TypeOKCtx
14
               TypeOKSerial
15
               copss \in [Replica \rightarrow SUBSET \ Cop]
16
17 F
    Init \triangleq
18
          \wedge InitInt
19
          \wedge InitCtx
20
          \land InitSerial
21
          \land copss = [r \in Replica \mapsto \{\}]
22
23
    RECURSIVE xForm(\_, \_)
    xForm(r, cop) \triangleq
25
         LET ctxDiff \stackrel{\Delta}{=} ds[r] \setminus cop.ctx Theorem : cop.ctx \subseteq ds[r]
26
               RECURSIVE xFormHelper(_, _, _)
27
                xFormHelper(coph, ctxDiffh, copssr) \stackrel{\triangle}{=} copssr: state space generated during transformation
28
                     IF ctxDiffh = \{\} THEN [xcop \mapsto coph, xcopss \mapsto copssr]
29
                      ELSE LET foph \stackrel{\triangle}{=} CHOOSE \ op \in ctxDiffh: the first op in serial
30
                                                  \forall opprime \in ctxDiffh \setminus \{op\} : tb(op, opprime, serial[r])
31
                                   fcophDict \stackrel{\Delta}{=} \{op \in copssr : op.oid = foph \land op.ctx = coph.ctx\}
32
                                   fcoph \stackrel{\triangle}{=} CHOOSE \ op \in fcophDict : TRUE \ THEOREM : Cardinality(fophDict) = 1
33
                                   xcoph \triangleq COT(coph, fcoph)
34
                                 xfcoph \triangleq COT(fcoph, coph)
35
                               IN xFormHelper(xcoph, ctxDiffh \setminus \{foph\}, copssr \cup \{xcoph, xfcoph\})
36
              xFormHelper(cop, ctxDiff, copss[r])
37
     Perform(r, cop) \triangleq
39
         LET xform \stackrel{\triangle}{=} xForm(r, cop) [xcop, xcopss]
40
                \land copss' = [copss \ EXCEPT \ ![r] = xform.xcopss \cup \{cop\}]
41
                \land SetNewAop(r, xform.xcop.op)
42
     ServerPerform(cop) \stackrel{\Delta}{=}
44
45
          \land Perform(Server, cop)
          \land Comm! SSendSame(ClientOf(cop), cop)
46
```

```
DoOp(c, op)
             \text{LET } cop \ \stackrel{\triangle}{=} \ [op \mapsto op, \ oid \mapsto [c \mapsto c, \ seq \mapsto cseq[c]], \ ctx \mapsto ds[c]] 
49
                     \land Perform(c, cop)
50
                     \land Comm! CSend(cop)
51
     Do(c) \triangleq
53
            \wedge DoInt(DoOp, c)
54
            \wedge DoCtx(c)
55
            \land DoSerial(c)
56
     Rev(c) \triangleq
58
            \land RevInt(Perform, c)
59
            \wedge RevCtx(c)
60
            \land RevSerial(c)
61
    SRev \triangleq
63
          \land SRevInt(ServerPerform)
64
          \land SRevCtx
65
          \land \ SRevSerial
66
67 H
    Next \triangleq
68
          \lor \exists c \in Client : Do(c) \lor Rev(c)
69
          \vee SRev
70
     Fairness \triangleq
72
         WF_{vars}(SRev \vee \exists c \in Client : Rev(c))
73
    Spec \stackrel{\triangle}{=} Init \wedge \Box [Next]_{vars} \wedge Fairness
76 F
     Compactness \triangleq
77
          Comm!EmptyChannel \Rightarrow Cardinality(Range(copss)) = 1
78
    Theorem Spec \Rightarrow Compactness
     \* Modification History
     \* Last modified Wed Jan 02 20:47:48 CST 2019 by hengxin
     \* Created Wed Dec 05 19:55:52 CST 2018 by hengxin
```