

```

1  |----- MODULE AbsJupiter -----|
   | Abstract Jupiter, inspired by the COT algorithm proposed by Sun and Sun. See their paper |
   | published on TPDS'2009. |
6  | EXTENDS JupiterSerial |
   |-----|
8  VARIABLES
9    copss   copss[r]: the state space (i.e., a set) of Cops maintained at replia r ∈ Replica
11  vars  $\triangleq$   $\langle \textit{intVars}, \textit{ctxVars}, \textit{serialVars}, \textit{copss} \rangle$ 
12  |-----|
13  TypeOK  $\triangleq$ 
14     $\wedge$  TypeOKInt
15     $\wedge$  TypeOKCtx
16     $\wedge$  TypeOKSerial
17     $\wedge$  Comm(Cop)! TypeOK
18     $\wedge$  copss ∈ [Replica → SUBSET Cop]
19  |-----|
20  Init  $\triangleq$ 
21     $\wedge$  InitInt
22     $\wedge$  InitCtx
23     $\wedge$  InitSerial
24     $\wedge$  Comm(Cop)! Init
25     $\wedge$  copss = [r ∈ Replica ↦ {}]
26  |-----|
27  RECURSIVE xForm(-, -)
28  xForm(cop, r)  $\triangleq$ 
29    LET ctxDiff  $\triangleq$  ds[r] \ cop.ctx THEOREM : cop.ctx ⊆ ds[r]
30    RECURSIVE xFormHelper(-, -, -)
31      xFormHelper(coph, ctxDiffh, copssr)  $\triangleq$  'h' stands for "helper"
32      IF ctxDiffh = {}
33      THEN  $\langle \textit{coph}, \textit{copssr} \rangle$ 
34      ELSE LET foph  $\triangleq$  CHOOSE op ∈ ctxDiffh : the first op (specifically, oid) in serial
35         $\forall \textit{opprime} \in \textit{ctxDiffh} :$ 
36          opprime ≠ op ⇒ tb(op, opprime, serial[r])
37          fcophDict  $\triangleq$  {op ∈ copssr : op.oid = foph ∧ op.ctx = coph.ctx}
38          fcoph  $\triangleq$  CHOOSE op ∈ fcophDict : TRUE THEOREM : Cardinality(fcophDict) = 1
39          cophx  $\triangleq$  COT(coph, fcoph)
40          fcophx  $\triangleq$  COT(fcoph, coph)
41          IN xFormHelper(cophx, ctxDiffh \ {foph}, copssr ∪ {cophx, fcophx})
42      IN xFormHelper(cop, ctxDiff, copss[r])
44  Perform(cop, r)  $\triangleq$ 
45    LET xform  $\triangleq$  xForm(cop, r)  $\langle \textit{xcop}, \textit{xcopss} \rangle$ 
46    xcop  $\triangleq$  xform[1]
47    xcopssr  $\triangleq$  xform[2]

```

```

48      IN     $\wedge state' = [state \text{ EXCEPT } ![r] = Apply(xcop.op, @)]$ 
49       $\wedge copss' = [copss \text{ EXCEPT } ![r] = xcopssr \cup \{cop\}]$ 
50  |-----|
      Client  $c \in Client$  issues an operation  $op$ .
54   $DoOp(c, op) \triangleq$   $op$ : the raw operation generated by the client  $c \in Client$ 
55       $\wedge LET \ cop \triangleq [op \mapsto op, oid \mapsto [c \mapsto c, seq \mapsto cseq'[c]], ctx \mapsto ds[c]]$ 
56      IN     $\wedge Perform(cop, c)$ 
57       $\wedge UpdateDS(c, cop)$ 
58       $\wedge Comm(Cop)!CSend(cop)$ 

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

65   $DoDel(c) \triangleq$ 
66       $\exists del \in \{op \in Del : op.pos \in 1 \dots Len(state[c])\} :$ 
67       $\wedge DoOp(c, del)$ 
68       $\wedge UNCHANGED \ chins$ 

70   $Do(c) \triangleq$ 
71       $\wedge DoCtx(c)$ 
72       $\wedge DoSerial(c)$ 
73       $\wedge \vee DoIns(c)$ 
74       $\vee DoDel(c)$ 
75  |-----|

76   $Rev(c) \triangleq$ 
77       $\wedge Comm(Cop)!CRev(c)$ 
78       $\wedge Perform(Head(cincoming[c]), c)$ 
79       $\wedge RevSerial(c)$ 
80       $\wedge RevCtx(c)$ 
81       $\wedge UNCHANGED \ chins$ 
82  |-----|

83   $SRev \triangleq$ 
84       $\wedge Comm(Cop)!SRev$ 
85       $\wedge LET \ cop \triangleq Head(sincoming)$ 
86      IN     $\wedge Perform(cop, Server)$ 
87       $\wedge Comm(Cop)!SSendSame(cop.oid.c, cop)$ 
88       $\wedge SRevSerial$ 
89       $\wedge SRevCtx$ 
90       $\wedge UNCHANGED \ chins$ 
91  |-----|

92   $Next \triangleq$ 
93       $\vee \exists c \in Client : Do(c) \vee Rev(c)$ 
94       $\vee SRev$ 

```

```

96 Fairness  $\triangleq$ 
97      $\text{WF}_{vars}(SRev \vee \exists c \in Client : Rev(c))$ 
99 Spec  $\triangleq$  Init  $\wedge \Box[Next]_{vars}$   $\wedge$  Fairness
100 |-----|
101 Compactness  $\triangleq$ 
102      $Comm(Cop)!EmptyChannel \Rightarrow Cardinality(Range(copss)) = 1$ 
104 THEOREM Spec  $\Rightarrow$  Compactness
105 |-----|
    \ * Modification History
    \ * Last modified Tue Dec 18 22:10:07 CST 2018 by hengxin
    \ * Created Wed Dec 05 19:55:52 CST 2018 by hengxin

```