```
- Module GraphStateSpace -
 1 1
    The graph representation of n-ary ordered state space and 2D state space used in CJupiter and
    XJupiter, respectively.
    EXTENDS JupiterCtx, GraphUtils
 7 |
     IsSS(G) \stackrel{\triangle}{=} A state space is a digraph with labeled edges.
           \wedge IsGraph(G) It is a digraph (represented by a record).
 9
           \land G.node \subseteq (SUBSET\ Oid) Each node represents a document state, i.e., a set of Oid.
10
           \land G.edge \subseteq [from: G.node, to: G.node, cop: Cop] Labeled with a Cop operation.
11
     EmptySS \stackrel{\triangle}{=} EmptyGraph
13
14
     Locate(cop, ss) \stackrel{\Delta}{=} Locate the node in state space ss that matches the context of cop.
15
         CHOOSE n \in ss.node : n = cop.ctx
16
    xForm(NextEdge(\_,\_,\_), r, cop, ss) \stackrel{\triangle}{=}
                                                           Transform cop with an operation sequence
18
         LET u \triangleq Locate(cop, ss)
                                                           in state space ss at replica r.
19
               v \triangleq u \cup \{cop.oid\}
20
               RECURSIVE xFormHelper(\_, \_, \_, \_)
21
                 xFormHelper(uh, vh, coph, xss) \stackrel{\Delta}{=}
22
                     IF uh = ds[r]
23
                      THEN [xcop \mapsto coph,
24
                                xss \mapsto xss, xss: eXtra ss created during transformation
25
                                lss \mapsto [node \mapsto \{vh\},
26
                                         edge \mapsto \{[from \mapsto uh, to \mapsto vh, cop \mapsto coph]\}]
27
                      ELSE LET e \stackrel{\Delta}{=} NextEdge(r, uh, ss)
28
                                     copprime \triangleq e.cop
29
                                     uprime \stackrel{\triangle}{=} e.to
30
                                     vprime \triangleq vh \cup \{copprime.oid\}
31
                                      coph2copprime \stackrel{\triangle}{=} COT(coph, copprime)
32
                                      copprime2coph \triangleq COT(copprime, coph)
33
                                      xFormHelper(uprime, vprime, coph2copprime,
                              IN
34
                                          xss \oplus [node \mapsto \{vprime\},\
35
                                                  edge \mapsto \{[from \mapsto vh, to \mapsto vprime,
36
                                                                 cop \mapsto copprime2coph,
37
                                                               [from \mapsto uprime, to \mapsto vprime,
38
                                                                 cop \mapsto coph2copprime[]])
39
                 xFormHelper(u, v, cop, [node \mapsto \{v\},
40
         IN
                                                 edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto cop]\}])
41
    xFormCopCops(cop, cops) \triangleq Transform cop against cops on state space.
43
         LET RECURSIVE xFormCopCopsSSHelper(\_, \_, \_)
44
                 xFormCopCopsSSHelper(coph, copsh, xss) \stackrel{\triangle}{=}
45
                     LET u \triangleq coph.ctx

v \triangleq u \cup \{coph.oid\}
46
47
                       uvSS \stackrel{\Delta}{=} [node \mapsto \{u, v\},
48
```

```
edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph]\}]
49
                            IF copsh = \langle \rangle THEN [xcop \mapsto coph,
50
                                                           xss \mapsto xss \oplus uvSS, lss \mapsto uvSS
51
                              ELSE LET copprimeh \stackrel{\triangle}{=} Head(copsh) uprime \stackrel{\triangle}{=} u \cup \{copprimeh.oid\}
52
53
                                                 \begin{array}{ll} vprime & \triangleq u \cup \{coph.oid, copprimeh.oid\} \end{array}
54
                                              coph2copprimeh \stackrel{\triangle}{=} COT(coph, copprimeh)
55
                                              copprimeh2coph \triangleq COT(copprimeh, coph)
56
                                              xFormCopCopsSSHelper(coph2copprimeh,
57
                                                   Tail(copsh),
                                                  xss \oplus [node \mapsto \{u, v\},
59
                                                           edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph],
60
                                                                        [from \mapsto u, to \mapsto uprime,
61
                                                                          cop \mapsto copprimeh],
62
                                                                        [from \mapsto v, to \mapsto vprime,
63
                                                                          cop \mapsto copprimeh2coph[\}]
64
                 xFormCopCopsSSHelper(cop, cops, EmptySS)
         IN
65
    xFormCopCopsShift(cop, cops, shift) \stackrel{\Delta}{=}
67
                                     shifting the first shift elements out of cops
68
69
         xFormCopCops(cop, SubSeq(cops, shift + 1, Len(cops)))
70
     \* Modification History
     \* Last modified Tue Feb 05 11:52:00 CST 2019 by hengxin
```

<sup>\\*</sup> Created Wed Dec 19 18:15:25 CST 2018 by hengxin