```
1  ---------- MODULE GraphStateSpace ----------
   The graph representation of n-ary ordered state space and 2D state space used in CJupiter and
   XJupiter, respectively.
6  EXTENDS JupiterCtx, GraphsUtil
7  ----------
8  IsSS(G) ≜  A state space is a digraph with labeled edges.
9      ∧ IsGraph(G)  It is a digraph (represented by a record).
10     ∧ G.node ⊆ (SUBSET Oid)  Each node represents a document state, i.e., a set of Oid.
11     ∧ G.edge ⊆ [from : G.node, to : G.node, cop : Cop]  Labeled with a Cop operation.

13 EmptySS ≜ EmptyGraph
14 ----------
15 Locate(cop, ss) ≜  Locate the node in state space ss that matches the context of cop.
16     CHOOSE n ∈ ss.node : n = cop.ctx

18 xForm(NextEdge(_, _, _), r, cop, ss) ≜  Transform cop with an operation sequence
19     LET u ≜ Locate(cop, ss)                in state space ss at replica r.
20         v ≜ u ∪ {cop.oid}
21         RECURSIVE xFormHelper(_, _, _, _)
22          xFormHelper(uh, vh, coph, xss) ≜
23              IF uh = ds[r]
24               THEN [xcop ↦ coph,
25                     xss ↦ xss,   xss: eXtra ss created during transformation
26                     lss ↦ [node ↦ {vh},
27                            edge ↦ {[from ↦ uh, to ↦ vh, cop ↦ coph]}]]
28               ELSE LET e ≜ NextEdge(r, uh, ss)
29                        copprime ≜ e.cop
30                        uprime ≜ e.to
31                        vprime ≜ vh ∪ {copprime.oid}
32                         coph2copprime ≜ COT(coph, copprime)
33                         copprime2coph ≜ COT(copprime, coph)
34                    IN   xFormHelper(uprime, vprime, coph2copprime,
35                           xss ⊕ [node ↦ {vprime},
36                                  edge ↦ {[from ↦ vh, to ↦ vprime,
37                                           cop ↦ copprime2coph],
38                                          [from ↦ uprime, to ↦ vprime,
39                                           cop ↦ coph2copprime]}])
40     IN   xFormHelper(u, v, cop, [node ↦ {v},
41                                  edge ↦ {[from ↦ u, to ↦ v, cop ↦ cop]}])

43 xFormCopCops(cop, cops) ≜  Transform cop against cops on state space.
44     LET RECURSIVE xFormCopCopsSSHelper(_, _, _)
45         xFormCopCopsSSHelper(coph, copsh, xss) ≜
46             LET u ≜ coph.ctx
47                 v ≜ u ∪ {coph.oid}
48              uvSS ≜ [node ↦ {u, v},
```

```
                        edge ↦ {[from ↦ u, to ↦ v, cop ↦ coph]}]
        IN   IF copsh = ⟨⟩ THEN [xcop ↦ coph,
                                 xss ↦ xss ⊕ uvSS, lss ↦ uvSS]
             ELSE LET copprimeh ≜ Head(copsh)
                      uprime ≜ u ∪ {copprimeh.oid}
                      vprime ≜ u ∪ {coph.oid, copprimeh.oid}
                  coph2copprimeh ≜ COT(coph, copprimeh)
                  copprimeh2coph ≜ COT(copprimeh, coph)
             IN   xFormCopCopsSSHelper(coph2copprimeh,
                      Tail(copsh),
                      xss ⊕ [node ↦ {u, v},
                             edge ↦ {[from ↦ u, to ↦ v, cop ↦ coph],
                                     [from ↦ u, to ↦ uprime,
                                      cop ↦ copprimeh],
                                     [from ↦ v, to ↦ vprime,
                                      cop ↦ copprimeh2coph]}])
    IN   xFormCopCopsSSHelper(cop, cops, EmptySS)

xFormCopCopsShift(cop, cops, shift) ≜
                    shifting the first shift elements out of cops
    xFormCopCops(cop, SubSeq(cops, shift + 1, Len(cops)))
```

* Modification History
* Last modified Sat *Jan* 19 16:32:38 *CST* 2019 by *hengxin*
* Created *Wed Dec* 19 18:15:25 *CST* 2018 by *hengxin*