```
MODULE StateSpace
The graph representation of n-ary ordered state space and 2D state space used in CJupiter and XJupiter, respectively.
EXTENDS JupiterCtx, GraphsUtil

IsSS(G) ≜ A state space is a digraph with labeled edges.
    ∧ IsGraph(G) It is a digraph (represented by a record).
    ∧ G.node ⊆ (SUBSET Oid) Each node represents a document state, i.e., a set of Oid.
    ∧ G.edge ⊆ [from : G.node, to : G.node, cop : Cop] Each edge is labeled with a Cop operation.

EmptySS ≜ EmptyGraph

Locate(cop, ss) ≜ Locate the (unique) node in state space ss that matches the context ctx of cop.
    CHOOSE n ∈ ss.node : n = cop.ctx

RECURSIVE ExtractCopSeq(_, _, _, _)
ExtractCopSeq(NextEdge(_, _, _), r, u, ss) ≜
    IF u = ds[r] THEN ⟨⟩
     ELSE LET e ≜ NextEdge(r, u, ss)
          IN ⟨e.cop⟩ ∘ ExtractCopSeq(NextEdge, r, e.to, ss)

xFormSS(cop, copprime) ≜ Transform cop against copprime on state space.
    LET u ≜ cop.ctx Return the extra state space.
        v ≜ u ∪ {cop.oid}
        uprime ≜ u ∪ {copprime.oid}
        vprime ≜ u ∪ {cop.oid, copprime.oid}
        cop2copprime ≜ COT(cop, copprime)
        copprime2cop ≜ COT(copprime, cop)
     IN [node ↦ {u, v, uprime, vprime},
         edge ↦ {[from ↦ u, to ↦ v, cop ↦ cop],
                 [from ↦ u, to ↦ uprime, cop ↦ copprime],
                 [from ↦ v, to ↦ vprime, cop ↦ copprime2cop],
                 [from ↦ uprime, to ↦ vprime, cop ↦ cop2copprime]}]

xFormCopCopsSS(cop, cops) ≜ Transform cop against cops (a sequence of Cop) on state space.
    LET RECURSIVE xFormCopCopsSSHelper(_, _, _) Return the extra state space.
        xFormCopCopsSSHelper(coph, copsh, xss) ≜ xss: the eXtra state space
            LET u ≜ coph.ctx
                v ≜ u ∪ {coph.oid}
             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)
```

* Modification History
* Last modified *Tue Jan* 08 13:47:19 *CST* 2019 by *hengxin*
* Created *Wed Dec* 19 18:15:25 *CST* 2018 by *hengxin*