```
┌──────────────────────────── MODULE StateSpace ────────────────────────────┐
```

The graph representation of n-ary ordered state spaces and $2D$ state spaces used in *CJupiter* and *XJupiter*, respectively.

```
EXTENDS JupiterCtx, GraphsUtil
├───────────────────────────────────────────────────────────────────────────┤
```

A state space is a directed graph with labeled edges. Each node is characterized by its context, a set of operations. Each edge is labeled with an operation.

```
IsSS(G) ≜
      ∧ IsGraph(G)
      ∧ G.node ⊆ (SUBSET Oid)
      ∧ G.edge ⊆ [from : G.node, to : G.node, cop : Cop]

EmptySS ≜ EmptyGraph
```

Locate the node in a state space that matches the context *ctx* of cop.

```
Locate(cop, ss) ≜ CHOOSE n ∈ ss.node : n = cop.ctx
```

Do transformation on state space. Return the extra state space.

```
xFormSS(cop, copprime) ≜
    LET u ≜ cop.ctx
        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]}]
```

Transform cop against cops (a sequence of cops) on state space. Return the extra state space.

```
xFormCopCopsSS(cop, cops) ≜
    LET RECURSIVE xFormCopCopsSSHelper(_, _, _)
        xFormCopCopsSSHelper(coph, copsh, xss) ≜     xss: the eXtra state space
            LET u  ≜ cop.ctx
                v ≜ u ∪ {cop.oid}
            IN  IF copsh = ⟨⟩
                THEN xss ⊕ [node ↦ {u, v},
                            edge ↦ {[from ↦ u, to ↦ v, cop ↦ cop]}]
                ELSE LET copprimeh ≜ Head(copsh)
                           uprime ≜ u ∪ {copprimeh.oid}
                           vprime ≜ u ∪ {coph.oid, copprimeh.oid}
                         coph2copprimeh  ≜ COT(coph, copprimeh)
                          copprimeh2coph ≜ COT(copprimeh, coph)
                     IN  xss ⊕ [node ↦ {u, v},
                                 edge ↦ {[from ↦ u, to ↦ v, cop ↦ cop],
                                               [from ↦ u, to ↦ uprime, cop ↦ copprimeh],
                                               [from ↦ v, to ↦ vprime, cop ↦ copprimeh2coph]}]
     IN  xFormCopCopsSSHelper(cop, cops, EmptySS)

 \* Modification History
 \* Last modified Sun Dec 30 15:21:14 CST 2018 by hengxin
 \* Created Wed Dec 19 18:15:25 CST 2018 by hengxin
```