

```

1  ┌────────────────────────── MODULE StateSpace ───────────────────┐
  The graph representation of  $n$ -ary ordered state spaces and 2D state spaces used in CJupiter and
  XJupiter, respectively.
6  EXTENDS JupiterCtx, GraphsUtil
7  └──────────────────────────────────────────────────────────────────┘

  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.

13 IsSS( $G$ )  $\triangleq$ 
14    $\wedge$  IsGraph( $G$ )
15    $\wedge$   $G.node \subseteq (\text{SUBSET } \textit{Oid})$ 
16    $\wedge$   $G.edge \subseteq [\textit{from} : G.node, \textit{to} : G.node, \textit{cop} : \textit{Cop}]$ 

18 EmptySS  $\triangleq$  EmptyGraph
  Locate the node in a state space that matches the context ctx of cop.

22 Locate(cop, ss)  $\triangleq$  CHOOSE  $n \in ss.node : n = cop.ctx$ 
  Do transformation on state space. Return the extra state space.

26 xFormSS(cop, copprime)  $\triangleq$ 
27   LET  $u \triangleq cop.ctx$ 
28    $v \triangleq u \cup \{cop.oid\}$ 
29    $uprime \triangleq u \cup \{copprime.oid\}$ 
30    $vprime \triangleq u \cup \{cop.oid, copprime.oid\}$ 
31    $cop2copprime \triangleq COT(cop, copprime)$ 
32    $copprime2cop \triangleq COT(copprime, cop)$ 
33   IN  $[node \mapsto \{u, v, uprime, vprime\},$ 
34        $edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto cop],$ 
35                    $[from \mapsto u, to \mapsto uprime, cop \mapsto copprime],$ 
36                    $[from \mapsto v, to \mapsto vprime, cop \mapsto copprime2cop],$ 
37                    $[from \mapsto uprime, to \mapsto vprime, cop \mapsto cop2copprime]]]$ 
  Transform cop against cops (a sequence of cops) on state space. Return the extra state space.

42 xFormCopCopsSS(cop, cops)  $\triangleq$ 
43   LET RECURSIVE xFormCopCopsSSHHelper( $-, -, -$ )
44   xFormCopCopsSSHHelper(coph, copsh, xss)  $\triangleq$  xss: the eXtra state space
45   LET  $u \triangleq coph.ctx$ 
46    $v \triangleq u \cup \{coph.oid\}$ 
47    $uvSS \triangleq [node \mapsto \{u, v\}, edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph]]]$ 
48   IN IF copsh =  $\langle \rangle$  THEN  $[lss \mapsto uvSS, xss \mapsto xss \oplus uvSS]$ 
49   ELSE LET copprimeh  $\triangleq$  Head(copsh)
50    $uprime \triangleq u \cup \{copprimeh.oid\}$ 
51    $vprime \triangleq u \cup \{coph.oid, copprimeh.oid\}$ 
52    $coph2copprimeh \triangleq COT(coph, copprimeh)$ 
53    $copprimeh2coph \triangleq COT(copprimeh, coph)$ 
54   IN xFormCopCopsSSHHelper(coph2copprimeh, Tail(copsh),
55    $xss \oplus [node \mapsto \{u, v\},$ 
56    $edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph],$ 

```

```

57                                     [from  $\mapsto u$ , to  $\mapsto uprime$ , cop  $\mapsto coprimeh$ ],
58                                     [from  $\mapsto v$ , to  $\mapsto vprime$ , cop  $\mapsto coprimeh2coph$ ]])
59     IN    xFormCopCopsSSHelper(cop, cops, EmptySS)
60 └──────────────────────────────────────────────────────────────────────────┘
    \ * Modification History
    \ * Last modified Wed Jan 02 22:01:40 CST 2019 by hengxin
    \ * Created Wed Dec 19 18:15:25 CST 2018 by hengxin

```