

```

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, GraphUtils
7  ┌──────────────────┐
8   $IsSS(G) \triangleq$  A state space is a digraph with labeled edges.
9     $\wedge IsGraph(G)$  It is a digraph (represented by a record).
10    $\wedge G.node \subseteq (SUBSET\ Oid)$  Each node represents a document state, i.e., a set of Oid.
11    $\wedge G.edge \subseteq [from : G.node, to : G.node, cop : Cop]$  Labeled with a Cop operation.
13  $EmptySS \triangleq EmptyGraph$ 
14 ┌──────────────────┐
15  $Locate(cop, ss) \triangleq$  Locate the node in state space  $ss$  that matches the context of  $cop$ .
16   CHOOSE  $n \in ss.node : n = cop.ctx$ 
18  $xForm(NextEdge(-, -, -), r, cop, ss) \triangleq$  Transform  $cop$  with an operation sequence
19   LET  $u \triangleq Locate(cop, ss)$  in state space  $ss$  at replica  $r$ .
20    $v \triangleq u \cup \{cop.oid\}$ 
21   RECURSIVE  $xFormHelper(-, -, -, -)$ 
22    $xFormHelper(uh, vh, coph, xss) \triangleq$ 
23     IF  $uh = ds[r]$ 
24     THEN  $[xcop \mapsto coph,$ 
25        $xss \mapsto xss,$   $xss: eXtra\ ss\ created\ during\ transformation$ 
26        $lss \mapsto [node \mapsto \{vh\},$ 
27          $edge \mapsto \{[from \mapsto uh, to \mapsto vh, cop \mapsto coph]\}]$ 
28     ELSE LET  $e \triangleq NextEdge(r, uh, ss)$ 
29        $copprime \triangleq e.cop$ 
30        $uprime \triangleq e.to$ 
31        $vprime \triangleq vh \cup \{copprime.oid\}$ 
32        $coph2copprime \triangleq COT(coph, copprime)$ 
33        $copprime2coph \triangleq COT(copprime, coph)$ 
34     IN  $xFormHelper(uprime, vprime, coph2copprime,$ 
35        $xss \oplus [node \mapsto \{vprime\},$ 
36        $edge \mapsto \{[from \mapsto vh, to \mapsto vprime,$ 
37          $cop \mapsto copprime2coph],$ 
38          $[from \mapsto uprime, to \mapsto vprime,$ 
39          $cop \mapsto coph2copprime]\}]$ 
40     IN  $xFormHelper(u, v, cop, [node \mapsto \{v\},$ 
41        $edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto cop]\}])$ 
43  $xFormCopCops(cop, cops) \triangleq$  Transform  $cop$  against  $cops$  on state space.
44   LET RECURSIVE  $xFormCopCopsSSHHelper(-, -, -)$ 
45    $xFormCopCopsSSHHelper(coph, copsh, xss) \triangleq$ 
46     LET  $u \triangleq coph.ctx$ 
47      $v \triangleq u \cup \{coph.oid\}$ 
48      $uvSS \triangleq [node \mapsto \{u, v\},$ 

```

