

```

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   $IsSS(G) \triangleq$  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    $\wedge G.edge \subseteq [from : G.node, to : G.node, cop : Cop]$  Each edge is 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    $xFormHelper(uh, vh, coph, xss) \triangleq$   $xss$ : eXtra  $ss$  created during transformation
23   IF  $uh = ds[r]$ 
24   THEN  $[xcop \mapsto coph,$ 
25      $xss \mapsto xss,$ 
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  $cops$  (a sequence of Cop) 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    $uvSS \triangleq [node \mapsto \{u, v\}, edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph]\}]$ 

```

```

49   IN    IF  $\text{coph} = \langle \rangle$  THEN [ $x\text{cop} \mapsto \text{coph}$ ,  $xss \mapsto xss \oplus uvSS$ ,  $lss \mapsto uvSS$ ]
      ELSE LET  $\text{copprimeh} \triangleq \text{Head}(\text{copsh})$ 
               $u_{\text{prime}} \triangleq u \cup \{\text{copprimeh.oid}\}$ 
               $v_{\text{prime}} \triangleq u \cup \{\text{coph.oid}, \text{copprimeh.oid}\}$ 
               $\text{coph2copprimeh} \triangleq \text{COT}(\text{coph}, \text{copprimeh})$ 
               $\text{copprimeh2coph} \triangleq \text{COT}(\text{copprimeh}, \text{coph})$ 
      IN     $x\text{FormCopCopsSSHHelper}(\text{coph2copprimeh}, \text{Tail}(\text{copsh}),$ 
               $xss \oplus [\text{node} \mapsto \{u, v\},$ 
                   $\text{edge} \mapsto \{[\text{from} \mapsto u, \text{to} \mapsto v, \text{cop} \mapsto \text{coph}],$ 
                       $[\text{from} \mapsto u, \text{to} \mapsto u_{\text{prime}}, \text{cop} \mapsto \text{copprimeh}],$ 
                       $[\text{from} \mapsto v, \text{to} \mapsto v_{\text{prime}}, \text{cop} \mapsto \text{copprimeh2coph}]]])$ 
60   IN     $x\text{FormCopCopsSSHHelper}(\text{cop}, \text{cops}, \text{EmptySS})$ 

62    $x\text{FormCopCopsShift}(\text{cop}, \text{cops}, \text{shift}) \triangleq$  shifting the first shift elements out of cops
63        $x\text{FormCopCops}(\text{cop}, \text{SubSeq}(\text{cops}, \text{shift} + 1, \text{Len}(\text{cops})))$ 
64   └──────────────────────────────────────────────────────────────────────────────────┘

/* Modification History
/* Last modified Sat Jan 19 10:49:59 CST 2019 by hengxin
/* Created Wed Dec 19 18:15:25 CST 2018 by hengxin

```