

```

1  |----- MODULE StateSpace -----|
   |The graph representation of  $n$ -ary ordered state space and 2D state space used in CJupiter and|
   |XJupiter, respectively.                |
6  |EXTENDS JupiterCtx, GraphsUtil      |
   |-----|
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 (unique) node in state space  $ss$  that matches the context  $ctx$  of  $cop$ .
16   CHOOSE  $n \in ss.node : n = cop.ctx$ 

18 RECURSIVE  $ExtractCopSeq(-, -, -)$ 
19  $ExtractCopSeq(u, ss, cur) \triangleq$ 
20   IF  $u = cur$  THEN  $\langle \rangle$ 
21   ELSE LET  $e \triangleq$  CHOOSE  $e \in ss.edge : e.from = u$ 
22   IN  $\langle e.cop \rangle \circ ExtractCopSeq(e.to, ss, cur)$ 

24  $xFormSS(cop, coprime) \triangleq$  Transform  $cop$  against  $coprime$  on state space.
25   LET  $u \triangleq cop.ctx$  Return the extra state space.
26    $v \triangleq u \cup \{cop.oid\}$ 
27    $uprime \triangleq u \cup \{coprime.oid\}$ 
28    $vprime \triangleq u \cup \{cop.oid, coprime.oid\}$ 
29    $cop2coprime \triangleq COT(cop, coprime)$ 
30    $coprime2cop \triangleq COT(coprime, cop)$ 
31   IN  $[node \mapsto \{u, v, uprime, vprime\},$ 
32      $edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto cop],$ 
33        $[from \mapsto u, to \mapsto uprime, cop \mapsto coprime],$ 
34        $[from \mapsto v, to \mapsto vprime, cop \mapsto coprime2cop],$ 
35        $[from \mapsto uprime, to \mapsto vprime, cop \mapsto cop2coprime]]]$ 

37  $xFormCopCopsSS(cop, cops) \triangleq$  Transform  $cop$  against  $cops$  (a sequence of Cop) on state space.
38   LET RECURSIVE  $xFormCopCopsSSHelper(-, -, -)$  Return the extra state space.
39    $xFormCopCopsSSHelper(coph, copsh, xss) \triangleq$   $xss$ : the eXtra state space
40   LET  $u \triangleq coph.ctx$ 
41    $v \triangleq u \cup \{coph.oid\}$ 
42    $uvSS \triangleq [node \mapsto \{u, v\}, edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph]]]$ 
43   IN IF  $copsh = \langle \rangle$  THEN  $[lss \mapsto uvSS, xss \mapsto xss \oplus uvSS]$ 
44   ELSE LET  $coprimeh \triangleq Head(copsh)$ 
45      $uprime \triangleq u \cup \{coprimeh.oid\}$ 
46      $vprime \triangleq u \cup \{coph.oid, coprimeh.oid\}$ 
47      $coph2coprimeh \triangleq COT(coph, coprimeh)$ 
48      $coprimeh2coph \triangleq COT(coprimeh, coph)$ 

```

```

49             IN   xFormCopCopsSSHelper(coph2copprimeh, Tail(copsh),
50               xss  $\oplus$  [node  $\mapsto$  {u, v},
51                 edge  $\mapsto$  {[from  $\mapsto$  u, to  $\mapsto$  v, cop  $\mapsto$  coph],
52                   [from  $\mapsto$  u, to  $\mapsto$  uprime, cop  $\mapsto$  copprimeh],
53                   [from  $\mapsto$  v, to  $\mapsto$  vprime, cop  $\mapsto$  copprimeh2coph]}])
54       IN   xFormCopCopsSSHelper(cop, cops, EmptySS)
55 |_____|
    \ * Modification History
    \ * Last modified Tue Jan 08 07:15:01 CST 2019 by hengxin
    \ * Created Wed Dec 19 18:15:25 CST 2018 by hengxin

```