
MODULE *GraphStateSpace*

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

EXTENDS *JupiterCtx*, *GraphsUtil*

$IsSS(G) \triangleq$ A state space is a digraph with labeled edges.
 $\wedge IsGraph(G)$ It is a digraph (represented by a record).
 $\wedge G.node \subseteq (SUBSET\ Oid)$ Each node represents a document state, *i.e.*, a set of *Oid*.
 $\wedge G.edge \subseteq [from : G.node, to : G.node, cop : Cop]$ Labeled with a *Cop* operation.

$EmptySS \triangleq EmptyGraph$

$Locate(cop, ss) \triangleq$ Locate the node in state space *ss* that matches the context of *cop*.
 CHOOSE $n \in ss.node : n = cop.ctx$

$xForm(NextEdge(-, -, -), r, cop, ss) \triangleq$ Transform *cop* with an operation sequence
 in state space *ss* at replica *r*.
 LET $u \triangleq Locate(cop, ss)$
 $v \triangleq u \cup \{cop.oid\}$
 RECURSIVE $xFormHelper(-, -, -, -)$
 $xFormHelper(uh, vh, coph, xss) \triangleq$
 IF $uh = ds[r]$
 THEN $[xcop \mapsto coph,$
 $xss \mapsto xss, \quad xss: eXtra\ ss\ created\ during\ transformation$
 $lss \mapsto [node \mapsto \{vh\},$
 $edge \mapsto \{[from \mapsto uh, to \mapsto vh, cop \mapsto coph]\}]$
 ELSE LET $e \triangleq NextEdge(r, uh, ss)$
 $copprime \triangleq e.cop$
 $uprime \triangleq e.to$
 $vprime \triangleq vh \cup \{copprime.oid\}$
 $coph2copprime \triangleq COT(coph, copprime)$
 $copprime2coph \triangleq COT(copprime, coph)$
 IN $xFormHelper(uprime, vprime, coph2copprime,$
 $xss \oplus [node \mapsto \{vprime\},$
 $edge \mapsto \{[from \mapsto vh, to \mapsto vprime,$
 $cop \mapsto copprime2coph],$
 $[from \mapsto uprime, to \mapsto vprime,$
 $cop \mapsto coph2copprime]\}]$
 IN $xFormHelper(u, v, cop, [node \mapsto \{v\},$
 $edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto cop]\}])$

$xFormCopCops(cop, cops) \triangleq$ Transform *cop* against *cops* on state space.
 LET RECURSIVE $xFormCopCopsSSHHelper(-, -, -)$
 $xFormCopCopsSSHHelper(coph, copsh, xss) \triangleq$
 LET $u \triangleq coph.ctx$
 $v \triangleq u \cup \{copsh.oid\}$
 $uvss \triangleq [node \mapsto \{u, v\},$

```

      edge  $\mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph]\}$ 
IN  IF  $copsh = \langle \rangle$  THEN  $[xcop \mapsto coph,$ 
       $xss \mapsto xss \oplus uvSS, lss \mapsto uvSS]$ 
ELSE LET  $copprimeh \triangleq Head(copsh)$ 
       $uprime \triangleq u \cup \{copprimeh.oid\}$ 
       $vprime \triangleq u \cup \{coph.oid, copprimeh.oid\}$ 
       $coph2copprimeh \triangleq COT(coph, copprimeh)$ 
       $copprimeh2coph \triangleq COT(copprimeh, coph)$ 
IN   $xFormCopCopsSSHelper(coph2copprimeh,$ 
       $Tail(copsh),$ 
       $xss \oplus [node \mapsto \{u, v\},$ 
       $edge \mapsto \{[from \mapsto u, to \mapsto v, cop \mapsto coph],$ 
       $[from \mapsto u, to \mapsto uprime,$ 
       $cop \mapsto copprimeh],$ 
       $[from \mapsto v, to \mapsto vprime,$ 
       $cop \mapsto copprimeh2coph]\})]$ 
IN   $xFormCopCopsSSHelper(cop, cops, EmptySS)$ 
 $xFormCopCopsShift(cop, cops, shift) \triangleq$ 
      shifting the first shift elements out of cops
       $xFormCopCops(cop, SubSeq(cops, shift + 1, Len(cops)))$ 

```

```

\ * Modification History
\ * Last modified Sat Jan 19 16:32:38 CST 2019 by anonymous
\ * Created Wed Dec 19 18:15:25 CST 2018 by anonymous

```