MODULE *DAG*

DiGraph operators taken from https://*github.com*/nano-o/TLA-Library/blob/master/*DiGraph.tla*

EXTENDS *FiniteSets*, *Naturals*, *Sequences*

The following two operators are from the Specifying Systems book, though I can't make *Path* work because of *Seq* being infinite and non-enumerable.

$Path(G) \triangleq$ The set of paths in G, where a path is represented as a sequence of nodes

$$\{p \in Seq(G.node) : \land\, p \neq \langle\rangle \\ \qquad \land\, \forall\, i \in 1 \,..\, (Len(p) - 1) : \langle p[i],\ p[i+1] \in G.edge\rangle\}$$

$AreConnected(m, n, G) \triangleq$ True if there is a path from m to n in G

$$\exists\, p \in Path(G) : (p[1] = m) \land (p[Len(p)] = n)$$

A digraph is a set of vertices and a set of edges, where an edge is a pair of vertices.

$Vertices(G) \triangleq G.node$
$Edges(G) \triangleq G.edge$
$IsDigraph(G) \triangleq Edges(G) \subseteq (Vertices(G) \times Vertices(G))$

Recursive implementation of $Dominates(v1, v2, G)$.

RECURSIVE $DominatesRec(\_, \_, \_, \_)$
$DominatesRec(v1, v2, G, acc) \triangleq$
$\quad \lor\, \langle v1, v2\rangle \in Edges(G)$
$\quad \lor\, \exists\, v \in Vertices(G) :$
$\qquad \land\, \neg v \in acc$
$\qquad \land\, \langle v1, v\rangle \in Edges(G)$
$\qquad \land\, DominatesRec(v, v2, G, acc \cup \{v1\})$

True when there exists a path from $v1$ to $v2$ in the graph G

$Dominates(v1, v2, G) \triangleq$
$\quad DominatesRec(v1, v2, G, \{\})$

All the paths of length smaller or equal to n in graph G

RECURSIVE $Paths(\_, \_)$
$Paths(n, G) \triangleq$
$\quad$ IF $n = 1$
$\quad$ THEN
$\qquad Edges(G)$
$\quad$ ELSE
$\qquad$ LET $nextVs(p) \triangleq$
$\qquad\qquad \{e[2] : e \in \{e \in Edges(G) : e[1] = p[Len(p)]\}\}$
$\qquad\quad nextPaths(p) \triangleq \{Append(p, v) : v \in nextVs(p)\}$
$\qquad$ IN

UNION $\{nextPaths(p) : p \in Paths(n-1,\ G)\}$
$\cup\, Paths(n-1,\ G)$