MODULE *SpanTreeTest*

The specification in this module is a modified version of the one in module *SpanTree* obtained by replacing the declared constant *Edges* with a variable of the same name that is set initially to any possible set of edges with nodes in *Nodes*. Thus, it can be used to test the algorithm of *SpanTree* on all possible graphs having a particular number of nodes.

EXTENDS *Integers*, *FiniteSets*, *Randomization*

CONSTANTS *Nodes*, *Root*, *MaxCardinality*

ASSUME $\land Root \in Nodes$
$\land MaxCardinality \in Nat$
$\land MaxCardinality \geq Cardinality(Nodes)$

VARIABLES *mom*, *dist*, *Edges*

$vars \triangleq \langle mom,\ dist,\ Edges \rangle$

$Nbrs(n) \triangleq \{m \in Nodes : \{m,\ n\} \in Edges\}$

$TypeOK \triangleq \land mom \in [Nodes \to Nodes]$
$\land dist \in [Nodes \to Nat]$
$\land \forall\, e \in Edges : (e \subseteq Nodes) \land (Cardinality(e) = 2)$

$Init \triangleq \land mom = [n \in Nodes \mapsto n]$
$\land dist = [n \in Nodes \mapsto \text{IF } n = Root \text{ THEN } 0 \text{ ELSE } MaxCardinality]$
$\land Edges \in \{E \in \text{SUBSET } (\text{SUBSET } Nodes) : \forall\, e \in E : Cardinality(e) = 2\}$

SUBSET S is the set of all subsets of a set S. Thus, this allows *Edges* to have as its initial value any set of sets of nodes containing exactly two nodes.

$Next \triangleq \exists\, n \in Nodes :$
$\exists\, m \in Nbrs(n) :$
$\land dist[m] < 1 + dist[n]$
$\land \exists\, d \in (dist[m] + 1) \,..\, (dist[n] - 1) :$
$\land dist' = [dist \text{ EXCEPT } ![n] = d]$
$\land mom' = [mom \text{ EXCEPT } ![n] = m]$
$\land Edges' = Edges$

$Spec \triangleq Init \land \Box[Next]_{vars} \land \text{WF}_{vars}(Next)$

$PostCondition \triangleq$
$\forall\, n \in Nodes :$
$\lor \land n = Root$
$\land dist[n] = 0$
$\land mom[n] = n$
$\lor \land dist[n] = MaxCardinality$
$\land mom[n] = n$
$\land \forall\, m \in Nbrs(n) : dist[m] = MaxCardinality$
$\lor \land dist[n] \in 1 \,..\, (MaxCardinality - 1)$
$\land mom[n] \in Nbrs(n)$

$\wedge\ dist[n] = dist[mom[n]] + 1$

$Safety \triangleq \Box((\neg \text{ENABLED } Next) \Rightarrow PostCondition)$

$Liveness \triangleq \Diamond PostCondition$

This took a few seconds to check for 4 nodes, and about 25 minutes for 5 nodes on my laptop. To compute the initial value of *Edges*, *TLC* has to compute all the elements of SUBSET (SUBSET *Nodes*) (the set of all subsets of the set of all sets of nodes) and then throw away all the elements of that set that don't consist entirely of sets having cardinality 2. For N nodes, SUBSET (SUBSET *Nodes*) contains $2\hat{}(2\hat{}N)$ elements.

* Modification History
* Last modified *Mon Jun* 17 05:43:38 *PDT* 2019 by *lamport*
* Created *Fri Jun* 14 03:07:58 *PDT* 2019 by *lamport*