MODULE *SpanTreeRandom*

The specification in this module is a modified version of the one in module *SpanTree* obtained by replacing the declared constant *Edges* with a defined constant that equals a randomly chosen set of edges joining the nodes in *Nodes*. Thus it can be used to test the algorithm of *SpanTree* on a randomly chosen node, making it easy to check the algorithm on a sequence of different graphs.

EXTENDS *Integers*, *FiniteSets*, *TLC*

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

$Edges \triangleq$
  UNION $\{\{\{n,\, m\} : m \in RandomElement(\text{SUBSET } (Nodes \setminus \{n\}))\} : n \in Nodes\}$

To understand this definition let's look at its subformulas, from the inside out.

- SUBSET $(Nodes \setminus \{n\})$ is the set of all subsets of the set $Nodes \setminus \{n\}$, which is the set of all nodes other than n.
- *RandomElement*(...) is a hack introduced in the *TLC* module. *TLC* computes its value to be a randomly chosen element in the set ... . This is hack because, in math, an expression has the same value whenever it's computed. The value of 2^{1/2} is the same next *Thursday* as it is today. Every mathematical expression *exp* satisfies $exp = exp$. However, *TLC* may evaluate

  $RandomElement(S) = RandomElement(S)$

  to equal FALSE if S is a set with more than 1 element, This is one of the few cases in which *TLC* does not obey the rules of math.
- $\{\{n,\, m\} : m \in RandomElement(\ldots)\}$ is the set of elements that equal the set $\{n,\, m\}$ for m some element of *RandomElement*(...) .
- UNION $\{\ldots : n \in Nodes\}$ is the union of all sets ... for n an element of *Nodes*. This expression makes sense if the expression equals a set that depends on the value of n.

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

VARIABLES *mom*, *dist*
$vars \triangleq \langle mom,\, dist \rangle$

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

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

$Init \triangleq \land\, mom = [n \in Nodes \mapsto n]$
  $\land\, dist = [n \in Nodes \mapsto \text{IF } n = Root \text{ THEN } 0 \text{ ELSE } MaxCardinality]$

$Next \triangleq \exists\, n \in Nodes :$
  $\exists\, m \in Nbrs(n) :$
    $\land\, dist[m] < 1 + dist[n]$
    $\land\, \exists\, d \in (dist[m] + 1) \,..\, (dist[n] - 1) :$

$$\begin{aligned} &\land\ dist' &&= [dist \text{ EXCEPT } ![n] &&= d] \\ &\land\ mom' &&= [mom \text{ EXCEPT } ![n] &&= m] \end{aligned}$$

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

$PostCondition \triangleq$

$\forall\, n \in Nodes :$

$\lor \land n = Root$
$\quad \land dist[n] = 0$
$\quad \land mom[n] = n$

$\lor \land dist[n] = MaxCardinality$
$\quad \land mom[n] = n$
$\quad \land \forall\, m \in Nbrs(n) : dist[m] = MaxCardinality$

$\lor \land dist[n] \in 1 \,..\, (MaxCardinality - 1)$
$\quad \land mom[n] \in Nbrs(n)$
$\quad \land dist[n] = dist[mom[n]] + 1$

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

$Liveness \triangleq \Diamond PostCondition$

Model *Model_1* has *TLC* check these correctness condition for a (randomly chosen) graph with six nodes. On a few tries, it took *TLC* an average of a little more than 30 seconds to do it.

---

* Modification History
* Last modified *Mon Jun* 17 05:39:15 *PDT* 2019 by *lamport*
* Created *Fri Jun* 14 03:07:58 *PDT* 2019 by *lamport*