## 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 $\mathit{Integers};\ \mathit{FiniteSets};\ \mathit{TLC}$

CONSTANTS $\mathit{Nodes};\ \mathit{Root};\ \mathit{MaxCardinality}$

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

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

- SUBSET $(\mathit{Nodes} \setminus \{n\})$ is the set of all subsets of the set $\mathit{Nodes} \setminus \{n\}$ , which is the set of all nodes other than $n$.
- $\mathit{RandomElement}(\ldots)$ is a hack introduced in the *TLC* module. *TLC* computes its value to be a randomly chosen element in the set $\ldots$ . 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 $\mathit{exp} = \mathit{exp}$. However, *TLC* may evaluate

  $$\mathit{RandomElement}(S) = \mathit{RandomElement}(S)$$

  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.
- $\{\{n,\ m\} : m \in \mathit{RandomElement}(\ldots)\}$ is the set of elements that equal the set $\{n,\ m\}$ for $m$ some element of $\mathit{RandomElement}(\ldots)$ .
- UNION $\{\ldots\ : n \in \mathit{Nodes}\}$ is the union of all sets $\ldots$ for $n$ an element of *Nodes*. This expression makes sense if the expression equals a set that depends on the value of $n$.

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

VARIABLES $\mathit{mom};\ \mathit{dist}$
$\mathit{vars} \triangleq \langle \mathit{mom};\ \mathit{dist} \rangle$

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

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

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

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

```
                  $\land\ dist' \;\; = [dist \text{ EXCEPT } ![n] \;\; = d]$
                  $\land\ mom' = [mom \text{ EXCEPT } ![n] = m]$

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

---

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

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

$Liveness \triangleq \Diamond PostCondition$
```

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*