Formal specification
Second assignment

Muhammad Reza Hidarian 95210551
Esmaeel Gholami 96210145

1 Finding minimum spanning tree for graph

Based on the definition of the second project, our aim here is to check whether or not a specific tree is minimum spanning tree for a specific graph. The main function is

Generic types

The only generic type here is vertex.

$$[Vertex]$$

Weighted interlocked graph

A weighted graph is a graph whose edges have a specific number assigned to them as weight and interlocked means that for every two different vertices there is a route connecting them. The following schema shows the identifiers and constraints for a weighted interlocked graph.

$$
\begin{array}{l}
\textbf{WI\_Graph} \\
\hline
v : \mathbb{F}\, Vertex \\
e : \mathbb{F}\, Vertex \rightarrow \mathbb{Z} \\
\hline
\forall x : \mathbb{F}\, Vertex \mid x \in \operatorname{dom} e \bullet \# x = 2 \land x \subseteq v \\
\forall v1, v2 : Vertex \mid v1 \in v \land v2 \in v \bullet \exists s : \operatorname{seq} Vertex \bullet \\
\quad \forall i : \mathbb{N}_1 \mid i < \# s \bullet \{s.1, s.i + 1\} \in \operatorname{dom} e \land \{v1, s.1\} \in \operatorname{dom} e \land \\
\qquad \{s.\# s, v2\} \in \operatorname{dom} e
\end{array}
$$

Weighted interlocked tree

A weighted tree is an interlocked graph whose edges have a specific number assigned to them as weight and the count of vertices is one more than the count of edges. The following schema shows the identifiers and constraints for a weighted interlocked tree.

$$
\begin{array}{l}
\textit{WI\_Tree} \\
\hline
v : \mathbb{F}\,Vertex \\
e : \mathbb{F}\,Vertex \to \mathbb{Z} \\
\hline
\forall\, x : \mathbb{F}\,Vertex \mid x \in \operatorname{dom} e \bullet \#\, x = 2 \wedge x \subseteq v \\
\forall\, v1, v2 : Vertex \mid v1 \in v \wedge v2 \in v \bullet \exists\, s : \operatorname{seq} Vertex \bullet \\
\qquad \forall\, i : \mathbb{N}_1 \mid i < \#\, s \bullet \{s.1, s.i + 1\} \in \operatorname{dom} e \wedge \{v1, s.1\} \in \operatorname{dom} e \wedge \\
\qquad\qquad \{s.\#\, s, v2\} \in \operatorname{dom} e \\
\#\, v = \#\operatorname{dom} e + 1
\end{array}
$$

Calculate weight function

The following function takes the edges of a graph and gives the summation of weights of all edges.

$$
\begin{array}{l}
weight : (\mathbb{F}\,vertex \to \mathbb{Z}) \to \mathbb{Z} \\
\hline
\forall\, x : \mathbb{F}\,vertex \to \mathbb{Z} \bullet \\
\#\, x = 0 \to weightx = 0 \\
\#\, x \neq 0 \to \exists\, e \in x \bullet weight\, x = e.2 + weight\,(x - e)
\end{array}
$$

check is spanning tree function

The following function takes a tree and a graph and checks whether or not the tree is the spanning tree of the graph.

$$
\begin{array}{l}
spanningTree\_\_ : WI\_Tree \times WI\_Graph \\
\hline
\forall\, wt : WI\_Tree, wg : WI\_Graph \bullet \\
spanningTree\, wt\, wg \Leftrightarrow wg.v = wt.v
\end{array}
$$

check is minimum spanning tree function

This is the main function and takes a tree and a graph and checks whether or not the tree is the minimum spanning tree of the graph. 1 meas positive and 0 means negative.

$$
\begin{array}{l}
minimumSpanningTree\_\_ : WI\_Tree \times WI\_Graph \\
\hline
\forall\, wt : WI\_Tree, wg : WI\_Graph \bullet \\
spanningTree\, wt\, wg \Leftrightarrow \\
\forall\, wt1 : WI\_Tree \mid spanning\, wt1\, wg \bullet weight\, wt \leq weight\, wt1
\end{array}
$$

2 Musical chair formal definition

Based on the description of the game there are two generic types:

$$[Player, Chair]$$

The schema of the game is as follows:

$$
\begin{array}{l}
\underline{\textit{MusicalChair}} \\
pl : \mathbb{F}\, Player \\
ch : \mathbb{F}\, Chair \\
occupiers : Chair \rightarrowtail Player \\
\hline
\#\, ch \geq 1 \\
\#\, pl = \#\, ch + 1 \\
\operatorname{dom} occupiers \subseteq ch \\
\operatorname{ran} occupiers \subseteq pl
\end{array}
$$

MusicPlay

The operation for when the music starts and keeps playing:

$$
\begin{array}{l}
\underline{\textit{MusicPlay}} \\
\Delta MusicalChair \\
\hline
pl' = pl \\
ch' = ch \\
occupiers' = \varnothing
\end{array}
$$

MusicStop

The operation for when the music stops:

$$
\begin{array}{l}
\underline{\textit{MusicStop}} \\
\Delta MusicalChair \\
out! : Player \\
\hline
ch' = ch \\
\operatorname{dom} occupiers' = ch' \\
out! \in pl - \operatorname{ran} occupiers'
\end{array}
$$

RemoveLoser

The operation for when the loser should be removed:

$$
\begin{array}{l}
\underline{\textit{RemoveLoser}} \\
\Delta MusicalChair \\
out? : Player \\
\hline
pl' = pl - \{out?\} \\
\exists\, chair : Chair \mid chair \in ch \bullet ch' = ch - \{chair\}
\end{array}
$$

Game

The game itself:

$$
\begin{array}{l}
MusicalChairInit == MusicalChairs';\ p? : \mathbb{F}\,Player;\ c? : \mathbb{F}\,Chair \mid \\
\quad \#\,c? \geq 1 \land \#\,p? = \#\,c? + 1 \land pl' = p? \land ch' = c?
\end{array}
$$

$$
\begin{array}{l}
GameFinished == MusicalChairs;\ winner! : Player \mid \\
\quad \#\,occupiers = 1 \land winner! \in \operatorname{ran} occupiers
\end{array}
$$

$$
\begin{array}{l}
PlayMusicalChairs == MusicalChairInit \mathbin{⨟} \\
\quad (MusicPlay \mathbin{⨟} MusicStop \mathbin{⨟} RemoveLoser[out!/out?])^n \mathbin{⨟} \\
\quad GameFinished \\
\qquad Where\, n = \#\,c? - 1
\end{array}
$$