MODULE *GalsIzhikevichOriginal*

EXTENDS *FiniteSets*, *Integers*

CONSTANTS *Neurons*, Total number of neurons
A tuple representing the set of in neighbours of each neuron
InNeighbours,
A tuple representing the set of out neighbours of each neuron
OutNeighbours,
MaxTime

The state variable is a function with domain the set of neurons and range a record with fieds: t (current time of neuron), p (number of pending fires), c (count of recieved messages)
VARIABLES *state*

Initialize the state of each neuron with $t = 0$, having 1 pending fire ($p = 1$) and not having received any messages ($c = 0$)

$$
\begin{array}{lll}
\mathit{GIInit} & \triangleq & \mathit{state} = [n \in 1 \,..\, \mathit{Neurons} \mapsto \\
& & \quad [\ t \mapsto 0,\ p \mapsto 1,\ c \mapsto 0\ ] \\
& &]
\end{array}
$$

$$
\begin{array}{lll}
\mathit{Next}(n) & \triangleq & \land\ \mathit{state}[n].t < \mathit{MaxTime} \\
& & \land\ \mathit{state}[n].p > 0 \\
& & \land\ \mathit{state}' = \\
& & \quad [a \in 1 \,..\, \mathit{Neurons} \mapsto \\
& & \quad\ \text{IF } a = n \text{ THEN} \\
& & \quad\quad [\ \mathit{state}[a] \text{ EXCEPT} \\
& & \quad\quad\quad !.t = @ + 1, \\
& & \quad\quad\quad !.p = @ - 1, \\
& & \quad\quad\quad !.c = 0 \\
& & \quad\quad] \\
& & \quad\ \text{ELSE IF } a \in \mathit{OutNeighbours}[n] \text{ THEN} \\
& & \quad\quad \text{Increment pending fires if count equal to the number of in neighbours} \\
& & \quad\quad \text{IF } \mathit{Cardinality}(\mathit{InNeighbours}[a]) = \mathit{state}[a].c + 1 \text{ THEN} \\
& & \quad\quad\quad [\ \mathit{state}[a] \text{ EXCEPT} \\
& & \quad\quad\quad\quad !.p = 1 + @, \\
& & \quad\quad\quad\quad !.c = 0 \\
& & \quad\quad\quad] \\
& & \quad\quad\ \text{ELSE} \\
& & \quad\quad\quad [\ \mathit{state}[a] \text{ EXCEPT } !.c = 1 + @\ ] \\
& & \quad\ \text{ELSE} \\
& & \quad\quad \mathit{state}[a] \\
& &]
\end{array}
$$

$GINext \triangleq \exists\, n \in 1 \,..\, Neurons : Next(n)$

$GISpec \triangleq GIInit \land \Box[GINext]_{\langle state \rangle}$

Check that the connections are correct

$$NeighbourOK \triangleq \forall\, n \in 1 \,..\, Neurons : \land \forall\, i \in InNeighbours[n] : n \in OutNeighbours[i] \land \forall\, o \in OutNeighbours[n] : n \in InNeighbours[o]$$

Check that the out neighbour is not more then 1 timestep ahead, need to receive the message of current time step before jumping the next one

$$TimeDiffOK \triangleq \forall\, n \in 1 \,..\, Neurons : \land \forall\, i \in InNeighbours[n] : \land state[n].t - state[i].t < 2 \land \forall\, o \in OutNeighbours[n] : \land state[n].t - state[o].t > -2$$

Check that the values of the state variables are correct

$$TypeOK \triangleq \land \forall\, n \in 1 \,..\, Neurons : \land state[n].t \leq MaxTime \land state[n].c \leq Cardinality(InNeighbours[n])$$
