MODULE *DiningPhilosophersWithRequests*

EXTENDS *Integers*, *FiniteSets*

CONSTANT N

VARIABLES *states*, *rightChopsticks*, *leftChopsticks*, *messages*

$$
\begin{aligned}
\mathit{TypeOK} \triangleq\ & \land (\forall\, n \in 1 \mathinner{..} N : \\
& \quad \land \mathit{states}[n] \in \{\text{“thinking”},\ \text{“waitingForRight”},\ \text{“waitingForLeft”},\ \text{“eating”}\} \\
& \quad \land \mathit{rightChopsticks}[n] \in \{\text{“available”},\ \text{“holding”},\ \text{“otherProcessHolding”}\} \\
& \quad \land \mathit{leftChopsticks}[n] \in \{\text{“available”},\ \text{“holding”},\ \text{“otherProcessHolding”}\}) \\
& \land (\forall\, m \in \mathit{messages} : \\
& \quad \land m.\mathit{to} \in 1 \mathinner{..} N \\
& \quad \land m.\mathit{from} \in 1 \mathinner{..} N \\
& \quad \land m.\mathit{type} \in \{\text{“rightChopstickRequest”},\ \text{“leftChopstickRequest”},\ \text{“rightChopstickReplyAccept”},
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Init} \triangleq\ & \land \mathit{states} = [n \in 1 \mathinner{..} N \mapsto \text{“thinking”}] \\
& \land \mathit{rightChopsticks} = [n \in 1 \mathinner{..} N \mapsto \text{“available”}] \\
& \land \mathit{leftChopsticks} = [n \in 1 \mathinner{..} N \mapsto \text{“available”}] \\
& \land \mathit{messages} = \{\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{rightIndex}(n) \triangleq \text{IF } n = N \text{ THEN } 1 \text{ ELSE } n + 1 \\
&\mathit{leftIndex}(n) \triangleq \text{IF } n = 1 \text{ THEN } N \text{ ELSE } n - 1
\end{aligned}
$$

$$
\mathit{requestRightChopstick}(n) \triangleq \mathit{messages}' = \mathit{messages} \cup \{[\mathit{from} \mapsto n,\ \mathit{to} \mapsto \mathit{rightIndex}(n),\ \mathit{type} \mapsto \text{“rightChopstick}
$$

$$
\mathit{requestLeftChopstick}(n) \triangleq \mathit{messages}' = \mathit{messages} \cup \{[\mathit{from} \mapsto n,\ \mathit{to} \mapsto \mathit{leftIndex}(n),\ \mathit{type} \mapsto \text{“leftChopstickReq}
$$

$$
\begin{aligned}
\mathit{tryToEat}(n) \triangleq\ & \land \mathit{states}[n] = \text{“thinking”} \\
& \land \mathit{requestLeftChopstick}(n) \\
& \land \mathit{states}' = [\mathit{states} \text{ EXCEPT } ![n] = \text{“waitingForLeft”}] \\
& \land \text{UNCHANGED } \langle \mathit{rightChopsticks},\ \mathit{leftChopsticks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{handleRightChopstickRequest}(n) \triangleq\ & \land (\exists\, m \in \mathit{messages} : \\
& \qquad \land m.\mathit{to} = n \\
& \qquad \land m.\mathit{type} = \text{“rightChopstickRequest”}) \\
& \land (\text{IF } (\land \mathit{leftChopsticks}[n] \in \{\text{“available”},\ \text{“otherProcessHolding”}\} \\
& \qquad \land \mathit{states}[n] \neq \text{“waitingForLeft”}) \\
& \quad \text{THEN } \mathit{messages}' = (\mathit{messages} \cup \{[\mathit{from} \mapsto n,\ \mathit{to} \mapsto \mathit{leftIndex}(n),\ \mathit{type} \mapsto \\
& \quad \text{ELSE } \mathit{messages}' = (\mathit{messages} \cup \{[\mathit{from} \mapsto n,\ \mathit{to} \mapsto \mathit{leftIndex}(n),\ \mathit{type} \mapsto \\
& \land \text{UNCHANGED } \langle \mathit{states},\ \mathit{rightChopsticks},\ \mathit{leftChopsticks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{handleLeftChopstickRequest}(n) \triangleq\ & \land (\exists\, m \in \mathit{messages} : \\
& \qquad \land m.\mathit{to} = n \\
& \qquad \land m.\mathit{type} = \text{“leftChopstickRequest”}) \\
& \land (\text{IF } (\land \mathit{rightChopsticks}[n] \in \{\text{“available”},\ \text{“otherProcessHolding”}\} \\
& \qquad \land \mathit{states}[n] \neq \text{“waitingForRight”})
\end{aligned}
$$

$\quad$ THEN $messages' = (messages \cup \{[from \mapsto n,\ to \mapsto rightIndex(n),\ type$
$\quad$ ELSE $messages' = (messages \cup \{[from \mapsto n,\ to \mapsto rightIndex(n),\ type$
$\wedge$ UNCHANGED $\langle states,\ rightChopsticks,\ leftChopsticks \rangle$

$handleRightChopstickReply(n) \triangleq$ FALSE

$handleLeftChopstickReply(n) \triangleq$ FALSE

$stopEating(n) \triangleq$ FALSE

$Next \triangleq \ \vee \exists\, n \in 1 \,..\, N :$
$\quad \vee\ tryToEat(n)$
$\quad \vee\ handleRightChopstickRequest(n)$
$\quad \vee\ handleLeftChopstickRequest(n)$
$\quad \vee\ handleRightChopstickReply(n)$
$\quad \vee\ handleLeftChopstickReply(n)$
$\quad \vee\ stopEating(n)$

$Stop \triangleq\ Cardinality(messages) = 6$

$AdjacentPeopleEating \triangleq\ \exists\, n \in 0 \,..\, N - 1 :$
$\quad \wedge\ states[rightIndex(n)] =$ "eating"
$\quad \wedge\ states[leftIndex(n)] =$ "eating"

* Modification History
* Last modified *Wed* May 04 22:19:01 *EDT* 2022 by *luke*
* Created *Wed* May 04 10:44:57 *EDT* 2022 by *luke*