

EXTENDS *Integers, FiniteSets, Sequences*

CONSTANT *N*

VARIABLES *states, rightChopsticks, leftChopsticks, messages*

$TypeOK \triangleq \wedge (\forall n \in 1 \dots N :$
 $\quad \wedge states[n] \in \{ \text{"thinking"}, \text{"waitingForRight"}, \text{"waitingForLeft"}, \text{"eating"} \}$
 $\quad \wedge rightChopsticks[n] \in \{ \text{"notHolding"}, \text{"holding"} \}$
 $\quad \wedge leftChopsticks[n] \in \{ \text{"notHolding"}, \text{"holding"} \})$
 $\wedge (\forall m \in messages :$
 $\quad \wedge m.to \in 1 \dots N$
 $\quad \wedge m.from \in 1 \dots N$
 $\quad \wedge m.type \in \{ \text{"rightChopstickRequest"}, \text{"leftChopstickRequest"}, \text{"rightChopstickReplyAccept"},$
 $\quad \text{"leftChopstickReplyAccept"}, \text{"rightChopstickReplyDeny"}, \text{"leftChopstickReplyDeny"} \})$

$Init \triangleq \wedge states = [n \in 1 \dots N \mapsto \text{"thinking"}]$
 $\quad \wedge rightChopsticks = [n \in 1 \dots N \mapsto \text{"notHolding"}]$
 $\quad \wedge leftChopsticks = [n \in 1 \dots N \mapsto \text{"notHolding"}]$
 $\quad \wedge messages = \{ \}$

$rightIndex(n) \triangleq \text{IF } n = N \text{ THEN } 1 \text{ ELSE } n + 1$
 $leftIndex(n) \triangleq \text{IF } n = 1 \text{ THEN } N \text{ ELSE } n - 1$

$tryToEat(n) \triangleq \text{LET } m \triangleq [from \mapsto n, to \mapsto leftIndex(n), type \mapsto \text{"leftChopstickRequest"}] \text{ IN}$
 $\quad \wedge states[n] = \text{"thinking"}$
 $\quad \wedge messages' = messages \cup \{m\}$
 $\quad \wedge states' = [states \text{ EXCEPT } ![n] = \text{"waitingForLeft"}]$
 $\quad \wedge \text{UNCHANGED } \langle rightChopsticks, leftChopsticks \rangle$

$acceptRightChopstickRequest(n) \triangleq$
 $\quad \text{LET } req \triangleq [from \mapsto leftIndex(n), to \mapsto n, type \mapsto \text{"rightChopstickRequest"}]$
 $\quad \quad resp \triangleq [from \mapsto n, to \mapsto leftIndex(n), type \mapsto \text{"rightChopstickReplyAccept"}] \text{ IN}$
 $\quad \wedge req \in messages$
 $\quad \wedge leftChopsticks[n] = \text{"notHolding"}$
 $\quad \wedge states[n] \neq \text{"waitingForLeft"}$
 $\quad \wedge messages' = (messages \cup \{resp\}) \setminus \{req\}$
 $\quad \wedge \text{UNCHANGED } \langle states, rightChopsticks, leftChopsticks \rangle$

$denyRightChopstickRequest(n) \triangleq$
 $\quad \text{LET } req \triangleq [from \mapsto leftIndex(n), to \mapsto n, type \mapsto \text{"rightChopstickRequest"}]$
 $\quad \quad resp \triangleq [from \mapsto n, to \mapsto leftIndex(n), type \mapsto \text{"rightChopstickReplyDeny"}] \text{ IN}$
 $\quad \wedge req \in messages$
 $\quad \wedge (leftChopsticks[n] = \text{"holding"} \vee states[n] = \text{"waitingForLeft"})$
 $\quad \wedge messages' = (messages \cup \{resp\}) \setminus \{req\}$
 $\quad \wedge \text{UNCHANGED } \langle states, rightChopsticks, leftChopsticks \rangle$

$acceptLeftChopstickRequest(n) \triangleq$
 $\text{LET } req \triangleq [from \mapsto rightIndex(n), to \mapsto n, type \mapsto \text{"leftChopstickRequest"}]$
 $\text{resp} \triangleq [from \mapsto n, to \mapsto rightIndex(n), type \mapsto \text{"leftChopstickReplyAccept"}] \text{IN}$
 $\wedge req \in messages$
 $\wedge rightChopsticks[n] = \text{"notHolding"}$
 $\wedge states[n] \neq \text{"waitingForRight"}$
 $\wedge messages' = (messages \cup \{resp\}) \setminus \{req\}$
 $\wedge \text{UNCHANGED } \langle states, rightChopsticks, leftChopsticks \rangle$

$denyLeftChopstickRequest(n) \triangleq$
 $\text{LET } req \triangleq [from \mapsto rightIndex(n), to \mapsto n, type \mapsto \text{"leftChopstickRequest"}]$
 $\text{resp} \triangleq [from \mapsto n, to \mapsto rightIndex(n), type \mapsto \text{"leftChopstickReplyDeny"}] \text{IN}$
 $\wedge req \in messages$
 $\wedge (rightChopsticks[n] = \text{"holding"} \vee states[n] = \text{"waitingForRight"})$
 $\wedge messages' = (messages \cup \{resp\}) \setminus \{req\}$
 $\wedge \text{UNCHANGED } \langle states, rightChopsticks, leftChopsticks \rangle$

$handleRightChopstickAccept(n) \triangleq$
 $\text{LET } reply \triangleq [from \mapsto rightIndex(n), to \mapsto n, type \mapsto \text{"rightChopstickReplyAccept"}] \text{IN}$
 $\wedge reply \in messages$
 $\wedge states[n] = \text{"waitingForRight"}$
 $\wedge rightChopsticks' = [rightChopsticks \text{ EXCEPT } ![n] = \text{"holding"}]$
 $\wedge states' = [states \text{ EXCEPT } ![n] = \text{"eating"}]$
 $\wedge messages' = messages \setminus \{reply\}$
 $\wedge \text{UNCHANGED } leftChopsticks$

$handleRightChopstickDeny(n) \triangleq$
 $\text{LET } reply \triangleq [from \mapsto rightIndex(n), to \mapsto n, type \mapsto \text{"rightChopstickReplyDeny"}] \text{IN}$
 $\wedge reply \in messages$
 $\wedge states[n] = \text{"waitingForRight"}$
 $\wedge states' = [states \text{ EXCEPT } ![n] = \text{"thinking"}]$
 $\wedge leftChopsticks' = [leftChopsticks \text{ EXCEPT } ![n] = \text{"notHolding"}]$
 $\wedge messages' = messages \setminus \{reply\}$
 $\wedge \text{UNCHANGED } rightChopsticks$

$handleLeftChopstickAccept(n) \triangleq$
 $\text{LET } reply \triangleq [from \mapsto leftIndex(n), to \mapsto n, type \mapsto \text{"leftChopstickReplyAccept"}]$
 $req \triangleq [from \mapsto n, to \mapsto rightIndex(n), type \mapsto \text{"rightChopstickRequest"}] \text{IN}$
 $\wedge reply \in messages$
 $\wedge states[n] = \text{"waitingForLeft"}$
 $\wedge leftChopsticks' = [leftChopsticks \text{ EXCEPT } ![n] = \text{"holding"}]$
 $\wedge states' = [states \text{ EXCEPT } ![n] = \text{"waitingForRight"}]$
 $\wedge messages' = (messages \cup \{req\}) \setminus \{reply\}$
 $\wedge \text{UNCHANGED } rightChopsticks$

$handleLeftChopstickDeny(n) \triangleq$

$$\begin{aligned}
& \text{LET } \textit{reply} \triangleq [\textit{from} \mapsto \textit{leftIndex}(n), \textit{to} \mapsto n, \textit{type} \mapsto \text{"leftChopstickReplyDeny"}] \text{IN} \\
& \wedge \textit{reply} \in \textit{messages} \\
& \wedge \textit{states}[n] = \text{"waitingForLeft"} \\
& \wedge \textit{states}' = [\textit{states} \text{ EXCEPT } ![n] = \text{"thinking"}] \\
& \wedge \textit{messages}' = \textit{messages} \setminus \{\textit{reply}\} \\
& \wedge \text{UNCHANGED } \langle \textit{rightChopsticks}, \textit{leftChopsticks} \rangle \\
\\
& \textit{stopEating}(n) \triangleq \\
& \quad \wedge \textit{states}[n] = \text{"eating"} \\
& \quad \wedge \textit{rightChopsticks}' = [\textit{rightChopsticks} \text{ EXCEPT } ![n] = \text{"notHolding"}] \\
& \quad \wedge \textit{leftChopsticks}' = [\textit{leftChopsticks} \text{ EXCEPT } ![n] = \text{"notHolding"}] \\
& \quad \wedge \textit{states}' = [\textit{states} \text{ EXCEPT } ![n] = \text{"thinking"}] \\
& \quad \wedge \text{UNCHANGED } \textit{messages} \\
\\
& \textit{Next} \triangleq \vee \exists n \in 1 \dots N : \\
& \quad \vee \textit{tryToEat}(n) \\
& \quad \vee \textit{acceptRightChopstickRequest}(n) \\
& \quad \vee \textit{denyRightChopstickRequest}(n) \\
& \quad \vee \textit{acceptLeftChopstickRequest}(n) \\
& \quad \vee \textit{denyLeftChopstickRequest}(n) \\
& \quad \vee \textit{handleRightChopstickAccept}(n) \\
& \quad \vee \textit{handleRightChopstickDeny}(n) \\
& \quad \vee \textit{handleLeftChopstickAccept}(n) \\
& \quad \vee \textit{handleLeftChopstickDeny}(n) \\
& \quad \vee \textit{stopEating}(n) \\
\\
& \textit{Stop} \triangleq \text{Len}(\text{SelectSeq}(\textit{states}, \text{LAMBDA } x : x = \text{"eating"})) = 3 \\
\\
& \textit{AdjacentPeopleEating} \triangleq \exists n \in 1 \dots N : \\
& \quad \wedge \textit{states}[n] = \text{"eating"} \\
& \quad \wedge \textit{states}[\textit{rightIndex}(n)] = \text{"eating"} \\
\\
& \textit{TwoPeopleHoldingChopstick} \triangleq \exists n \in 1 \dots N : \\
& \quad \wedge \textit{rightChopsticks}[n] = \text{"holding"} \\
& \quad \wedge \textit{leftChopsticks}[\textit{rightIndex}(n)] = \text{"holding"}
\end{aligned}$$

\ * Modification History
\ * Last modified Tue May 10 12:42:02 EDT 2022 by luke
\ * Created Wed May 04 10:44:57 EDT 2022 by luke