

Once upon a time a farmer went to a market and purchased a wolf, a goat, and a cabbage. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and a single one of his purchases: the wolf, the goat, or the cabbage.

If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage.

The farmer's challenge was to carry himself and his purchases to the far bank of the river, leaving each purchase intact. How did he do it?

EXTENDS *Sequences*

leftB - river bank where farmer came, *rightB* - far river bank, *boatPos* - boat position

VARIABLES *leftB*, *rightB*, *boatPos*

vars $\triangleq \langle \textit{leftB}, \textit{rightB}, \textit{boatPos} \rangle$

TypeOK \triangleq

$\wedge \textit{boatPos} \in \{ \text{"l"}, \text{"r"} \}$
 $\wedge \forall x \in \textit{leftB} : x \in \{ \text{"w"}, \text{"g"}, \text{"c"} \}$
 $\wedge \forall x \in \textit{rightB} : x \in \{ \text{"w"}, \text{"g"}, \text{"c"} \}$

Init \triangleq

$\wedge \textit{leftB} = \{ \text{"w"}, \text{"g"}, \text{"c"} \}$
 $\wedge \textit{rightB} = \{ \}$
 $\wedge \textit{boatPos} = \text{"l"}$

CanCross(place) \triangleq

$\wedge \neg(\{ \text{"w"}, \text{"g"} \} \subseteq \textit{place})$
 $\wedge \neg(\{ \text{"g"}, \text{"c"} \} \subseteq \textit{place})$

GotoRight \triangleq

$\wedge \textit{boatPos} = \text{"l"}$
 $\wedge \textit{CanCross}(\textit{leftB})$
 $\wedge \textit{boatPos}' = \text{"r"}$
 $\wedge \text{UNCHANGED } \langle \textit{leftB}, \textit{rightB} \rangle$

GotoLeft \triangleq

$\wedge \textit{boatPos} = \text{"r"}$
 $\wedge \textit{CanCross}(\textit{rightB})$
 $\wedge \textit{boatPos}' = \text{"l"}$
 $\wedge \text{UNCHANGED } \langle \textit{leftB}, \textit{rightB} \rangle$

Move(x, from, to) \triangleq

$\wedge \textit{from}' = \textit{from} \setminus \{x\}$
 $\wedge \textit{to}' = \textit{to} \cup \{x\}$

MovetoRight(x) \triangleq

$\wedge \textit{boatPos} = \text{"l"}$
 $\wedge \textit{CanCross}(\textit{leftB} \setminus \{x\})$

$$\begin{aligned}
& \wedge \textit{boatPos}' = \textit{"r"} \\
& \wedge \textit{Move}(x, \textit{leftB}, \textit{rightB}) \\
\textit{MovetoLeft}(x) & \triangleq \\
& \wedge \textit{boatPos} = \textit{"r"} \\
& \wedge \textit{CanCross}(\textit{rightB} \setminus \{x\}) \\
& \wedge \textit{boatPos}' = \textit{"l"} \\
& \wedge \textit{Move}(x, \textit{rightB}, \textit{leftB}) \\
\textit{Next} & \triangleq \\
& \vee \exists x \in \textit{leftB} : \textit{MovetoRight}(x) \\
& \vee \textit{GotoRight} \\
& \vee \exists x \in \textit{rightB} : \textit{MovetoLeft}(x) \\
& \vee \textit{GotoLeft}
\end{aligned}$$

$$\textit{Spec} \triangleq \textit{Init} \wedge \Box[\textit{Next}]_{\langle \textit{vars} \rangle}$$

THEOREM $\textit{Spec} \Rightarrow \textit{TypeOK}$

\ * Modification History
\ * Last modified *Fri Sep 10 15:23:16 MSK 2021* by *s.elin*
\ * Created *Sun Sep 05 09:59:38 MSK 2021* by *s.elin*