MODULE *FarmerCrossesRiver*

EXTENDS *Integers*, *FiniteSets*

A farmer stands in front of a large river. It has no bridge. There is a fence on the other side.
He wants to bring over a wolf, a goat and a cabbage in his rowing boat. But he can only take one thing per trip!
ATTENTION: If the farmer is absent, the wolf can eat the goat and the goat can eat the cabbage.

VARIABLES *carriage_on_side*, *boat*, *boat_side*, *last_carriage*
$vars \triangleq \langle carriage\_on\_side, boat, boat\_side, last\_carriage \rangle$

$goods\_to\_transport \triangleq \{\text{"goat"}, \text{"wulf"}, \text{"cabbage"}\}$
$boat\_docks \triangleq \{\text{"start"}, \text{"end"}\}$

---

$Init \triangleq$
$\land\ carriage\_on\_side = [start \mapsto goods\_to\_transport,\ end \mapsto \{\}]$
$\land\ boat = \{\}$
$\land\ boat\_side = \text{"start"}$
$\land\ last\_carriage = \text{"NULL"}$

---

$TypeOK \triangleq$
$\land\quad carriage\_on\_side \in [boat\_docks \to \text{SUBSET } goods\_to\_transport]$
$\land\quad boat \subseteq goods\_to\_transport$
$\land\quad Cardinality(boat) \leq 1$
$\land\quad boat\_side \in boat\_docks$
$\land\quad last\_carriage \in \{\text{"NULL"}\} \cup goods\_to\_transport$

---

$Safe(side) \triangleq$
$\lor\ \land (\{\text{"goat"}, \text{"wulf"}\} \subseteq side) = \text{FALSE}$
$\quad\ \land (\{\text{"goat"}, \text{"cabbage"}\} \subseteq side) = \text{FALSE}$
$\lor\ goods\_to\_transport \subseteq side$

$Consistent \triangleq$
LET $all\_participants \triangleq (carriage\_on\_side[\text{"start"}] \cup carriage\_on\_side[\text{"end"}] \cup boat)$
IN $\land\ all\_participants \setminus goods\_to\_transport = \{\}$
$\quad\ \land\ Cardinality(all\_participants) = 3$

---

$OtherSide(bs) \triangleq \text{CHOOSE } s \in boat\_docks : s \neq bs$

$BoatIsEmpty \triangleq Cardinality(boat) = 0$
$BoatIsLoaded \triangleq Cardinality(boat) = 1$

$ThisSideCarriage \triangleq carriage\_on\_side[boat\_side]$

---

$UpdateCarriageStatus(new\_this\_side) \triangleq$
I would love to write something like this, but do not know how to use
variable as string value for key in struct, *TLC* module *toString* does not help
LET $not\_boat\_side \triangleq OtherSide(boat\_side)$
IN $carriage\_on\_side' = [boat\_side \mapsto new\_this\_side, not\_boat\_side \mapsto carriage\_on\_side[\text{“end”}]]$
IF $boat\_side = \text{“start”}$
THEN $carriage\_on\_side' = [start \mapsto new\_this\_side, end \mapsto carriage\_on\_side[\text{“end”}]]$
ELSE $carriage\_on\_side' = [end \mapsto new\_this\_side, start \mapsto carriage\_on\_side[\text{“start”}]]$

$UpdateBoatIfSafe(new\_this\_side, new\_boat) \triangleq$
$\land\ Safe(new\_this\_side)$
$\land\ boat' = new\_boat$
$\land$ UNCHANGED $boat\_side$
$\land\ UpdateCarriageStatus(new\_this\_side)$

---

$ChangeBoatContent(participant) \triangleq$
$\land\ participant \neq last\_carriage$
$\land$ LET $new\_this\_side \triangleq (ThisSideCarriage \setminus \{participant\}) \cup boat$
$new\_boat \triangleq \{participant\}$
IN $\land\ last\_carriage' = participant$
$\land\ UpdateBoatIfSafe(new\_this\_side, new\_boat)$

$UnloadBoat \triangleq$
$\land\ BoatIsLoaded$
$\land$ LET $new\_this\_side \triangleq ThisSideCarriage \cup boat$
$new\_boat \triangleq \{\}$
IN $\land\ last\_carriage' = last\_carriage$
$\land\ UpdateBoatIfSafe(new\_this\_side, new\_boat)$

$RowOverToOtherSide \triangleq$
$\land\ boat\_side' = OtherSide(boat\_side)$
$\land$ UNCHANGED $\langle carriage\_on\_side, boat, last\_carriage\rangle$

$Transport \triangleq$
$\lor\ \exists\, participant \in ThisSideCarriage : ChangeBoatContent(participant)$
$\lor\ UnloadBoat$
$\lor\ RowOverToOtherSide$

---

$Next \triangleq Consistent \land Transport$

$Spec \triangleq Init \land \quad \Box[Next]_{vars}$

THEOREM $Spec \Rightarrow \Box TypeOK$

---

Ensure we get a *Stacktrace* containing the Solution, set this as Invariant

$NoSolution \triangleq Cardinality(carriage\_on\_side[\text{“end”}]) < 3$

---

* Modification History
* Last modified *Tue Mar* 05 16:49:27 *CET* 2024 by *jeujeus*
* Last modified *Wed Feb* 28 08:17:55 *CET* 2024 by *JUFIGGE*
* Last modified *Tue Feb* 27 17:34:16 *CET* 2024 by *JeuJeus*
* Created *Mon Feb* 26 12:41:56 *CET* 2024 by *JeuJeus*