MODULE AB

EXTENDS *Integers*, *Sequences*

CONSTANT *Data*

We first define $Remove(i, seq)$ to be the sequence obtained by removing element number i from sequence seq.

$$Remove(i, seq) \triangleq [j \in 1 \,..\, (Len(seq) - 1) \mapsto \text{IF } j < i \text{ THEN } seq[j] \text{ ELSE } seq[j+1]]$$

VARIABLES $AVar$, $BVar$, The same as in module *ABSpec*
$AtoB$, The sequence of data messages in transit from sender to receiver.
$BtoA$ The sequence of ack messages in transit from receiver to sender.
Messages are sent by appending them to the end of the sequence.
and received by removing them from the head of the sequence.

$$vars \triangleq \langle AVar, BVar, AtoB, BtoA \rangle$$

$$\begin{aligned} TypeOK \triangleq\ & \land AVar \in Data \times \{0, 1\} \\ & \land BVar \in Data \times \{0, 1\} \\ & \land AtoB \in Seq(Data \times \{0, 1\}) \\ & \land BtoA \in Seq(\{0, 1\}) \end{aligned}$$

$$\begin{aligned} Init \triangleq\ & \land AVar \in Data \times \{1\} \\ & \land BVar = AVar \\ & \land AtoB = \langle\rangle \\ & \land BtoA = \langle\rangle \end{aligned}$$

The action of the sender sending a data message by appending $AVar$ to the end of the message queue $AtoB$. It will keep sending the same message until it receives an acknowledgment for it from the receiver.

$$\begin{aligned} ASnd \triangleq\ & \land AtoB' = Append(AtoB, AVar) \\ & \land \text{UNCHANGED } \langle AVar, BtoA, BVar \rangle \end{aligned}$$

The action of the sender receiving an ack message. If that ack is for the value it is sending, then it chooses another message to send and sets $AVar$ to that message. If the ack is for the previous value it sent, it ignores the message. In either case, it removes the message from $BtoA$.

$$\begin{aligned} ARcv \triangleq\ & \land BtoA \neq \langle\rangle \\ & \land \text{IF } Head(BtoA) = AVar[2] \\ & \quad \text{THEN } \exists\, d \in Data : AVar' = \langle d, 1 - AVar[2] \rangle \\ & \quad \text{ELSE } AVar' = AVar \\ & \land BtoA' = Tail(BtoA) \\ & \land \text{UNCHANGED } \langle AtoB, BVar \rangle \end{aligned}$$

The action of the receiver sending an acknowledgment message for the last data item it received.

$$\begin{aligned} BSnd \triangleq\ & \land BtoA' = Append(BtoA, BVar[2]) \\ & \land \text{UNCHANGED } \langle AVar, BVar, AtoB \rangle \end{aligned}$$

The action of the receiver receiving a data message. It sets $BVar$ to that message if it's not for the data item it has already received.

$$
\begin{array}{l}
BRcv \triangleq \land\ AtoB \neq \langle\rangle \\
\qquad \land \text{IF } Head(AtoB)[2] \neq BVar[2] \\
\qquad\qquad \text{THEN } BVar' = Head(AtoB) \\
\qquad\qquad \text{ELSE } BVar' = BVar \\
\qquad \land\ AtoB' = Tail(AtoB) \\
\qquad \land \text{UNCHANGED } \langle AVar,\ BtoA\rangle
\end{array}
$$

LoseMsg is the action that removes an arbitrary message from queue $AtoB$ or $BtoA$.

$$
\begin{array}{l}
LoseMsg \triangleq \land \lor \land \exists\, i \in 1 \,..\, Len(AtoB) : \\
\qquad\qquad\qquad\qquad AtoB' = Remove(i,\ AtoB) \\
\qquad\qquad\qquad \land\ BtoA' = BtoA \\
\qquad\qquad \lor \land \exists\, i \in 1 \,..\, Len(BtoA) : \\
\qquad\qquad\qquad\qquad BtoA' = Remove(i,\ BtoA) \\
\qquad\qquad\qquad \land\ AtoB' = AtoB \\
\qquad\qquad \land \text{UNCHANGED } \langle AVar,\ BVar\rangle
\end{array}
$$

$$Next \triangleq ASnd \lor ARcv \lor BSnd \lor BRcv \lor LoseMsg$$

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

$$ABS \triangleq \text{INSTANCE } ABSpec$$

$$\text{THEOREM } Spec \Rightarrow ABS\,!\,Spec$$

FairSpec is *Spec* with fairness conditions added.

$$
\begin{array}{l}
FairSpec \triangleq Spec \ \land\ \mathrm{SF}_{vars}(ARcv) \ \land \mathrm{SF}_{vars}(BRcv) \land \\
\qquad\qquad\qquad\qquad \mathrm{WF}_{vars}(ASnd) \land \mathrm{WF}_{vars}(BSnd)
\end{array}
$$

* Modification History
* Last modified *Thu* May 17 07:48:10 *CST* 2018 by *tangruize*
* Last modified *Wed Dec* 27 13:29:51 *PST* 2017 by *lamport*
* Created *Wed Mar* 25 11:53:40 *PDT* 2015 by *lamport*