MODULE *LIFO_Interface*

EXTENDS *Naturals*, *Sequences*
CONSTANT *QueueSize*, *Message*
VARIABLES *q*, *in*, *out*

$InChan \triangleq$ INSTANCE *Channel* WITH *Data* $\leftarrow$ *Message*, *chan* $\leftarrow$ *in*
$OutChan \triangleq$ INSTANCE *Channel* WITH *Data* $\leftarrow$ *Message*, *chan* $\leftarrow$ *out*

Generalized sender/reciever interface - The *Send* and *Rcv* methods and a queue is specified, while specifics about how its done is omitted

$$\begin{aligned} Init \triangleq\ & \land\ InChan!Init \\ & \land\ OutChan!Init \\ & \land\ q = \langle\rangle \end{aligned}$$

$$\begin{aligned} TypeInvariant \triangleq\ & \land\ InChan!TypeInvariant \\ & \land\ OutChan!TypeInvariant \\ & \land\ q \in Seq(Message) \\ & \land\ QueueSize > 0 \\ & \land\ QueueSize \in Nat \\ & \land\ Len(q) \leq QueueSize \end{aligned}$$

$$\begin{aligned} Send(msg) \triangleq\ & \land\ InChan!Send(msg) \quad \text{Send } msg \text{ on channel } in \text{ .} \\ & \land\ \text{UNCHANGED } \langle out,\ q\rangle \\ & \land\ Len(q) < QueueSize \end{aligned}$$

$$\begin{aligned} Rcv \triangleq\ & \land\ OutChan!Rcv \quad \text{Receive message from channel } out \text{ .} \\ & \land\ \text{UNCHANGED } \langle in,\ q\rangle \end{aligned}$$

$$\begin{aligned} INext \triangleq\ & \lor\ \exists\, msg \in Message : Send(msg) \\ & \lor\ Rcv \end{aligned}$$

Rcv should eventually be called if $Send(msg)$ has been enabled

$$Liveness \triangleq \exists\, msg \in Message : \mathrm{WF}_{\langle in,\ out,\ q\rangle}(Send(msg) \lor Rcv)$$