MODULE *InnerLIFO*

EXTENDS *Naturals*, *Sequences*
CONSTANT *Message*, *BufSize*
VARIABLES *in*, *out*, *q*
$InChan \triangleq$ INSTANCE *Channel* WITH *Data* $\leftarrow$ *Message*, *chan* $\leftarrow$ *in*
$OutChan \triangleq$ INSTANCE *Channel* WITH *Data* $\leftarrow$ *Message*, *chan* $\leftarrow$ *out*

Functions and operators

Wraps $Len(s)$ operator into a function
$MyLen \triangleq [s \in Seq(Nat) \mapsto Len(s)]$

Simple increment function
$inc \triangleq [i \in Nat \mapsto i + 1]$

Simple increment operator
$OpInc(i) \triangleq i + 1$

Recursive implementation of the length of a sequence
$RecursiveLen[s \in Seq(Nat)] \triangleq$ IF $s = \langle\rangle$ THEN 0 ELSE $1 + RecursiveLen[Tail(s)]$

The model cannot be run when using this definition
$RecLen \triangleq$ CHOOSE $RecLen$:
$RecLen = [s \in Seq(Nat) \mapsto$ IF $s = \langle\rangle$ THEN 0 ELSE $1 + RecLen[Tail(s)]]$

Recursive implementation of a factorial function
$fact[n \in Nat] \triangleq$ IF $n = 0$ THEN 1 ELSE $n * fact[n - 1]$

Recursive function for finding the n'th fibonacci number
$fib[n \in Nat] \triangleq$ IF $n \leq 1$ THEN n ELSE $fib[n - 1] + fib[n - 2]$

$$Init \triangleq \land InChan!Init \land OutChan!Init \land q = \langle\rangle$$

$$TypeInvariant \triangleq \land InChan!TypeInvariant \land OutChan!TypeInvariant \land q \in Seq(Message) \land BufSize \in Nat \land RecursiveLen[q] \leq BufSize$$

$SSend(msg) \triangleq$ Send *msg* on channel *in* .
$\land InChan!Send(msg)$
$\land$ UNCHANGED $\langle out, q\rangle$
$\land RecursiveLen[q] < BufSize$

$BufRcv \triangleq$ Receive message from channel *in* .
insert *val* at the head of *q*.
$\land InChan!Receive$
$\land q' = \langle in.val\rangle \circ q$

$\land$ UNCHANGED out

$BufSend \triangleq$ $\land\ RecursiveLen[q] > 0$ — Enabled only if q is nonempty.
$\land\ OutChan!Send(Head(q))$ — Send $Head(q)$ on channel out
$\land\ q' = Tail(q)$ — and remove it from q.
$\land$ UNCHANGED in

$RRcv \triangleq$ $\land\ OutChan!Receive$ — Receive message from channel out .
$\land$ UNCHANGED $\langle in,\ q\rangle$

$Next \triangleq$ $\lor \exists\, msg \in Message : SSend(msg)$
$\lor\ BufRcv$
$\lor\ BufSend$
$\lor\ RRcv$

$Spec \triangleq Init \land \Box[Next]_{\langle in,\ out,\ q\rangle}$

THEOREM $Spec \Rightarrow \Box\, TypeInvariant$

\ * Modification History
\ * Last modified *Wed Mar* 14 11:56:59 *CET* 2018 by *jacob*
\ * Created *Mon Feb* 12 14:42:22 *CET* 2018 by *jacob*