MODULE *HourClockChannel*

1. (a) Write a specification of an hour-clock that sends the time to the environment over a channel *chan*. The specification should make use of the definitions from the *Channel* and *HourClock* modules by incorporating them with an EXTENDS statement. Write two versions of the specification.

   Version 1: The clock can tick at any time.

   Version 2: The clock cannot tick between sending a value on *chan* and the receipt of that value by the environment.

   Include type invariants and use *TLC* to check them.

   (b) Use *TLC* to check that the Version 1 specification implements the *Channel* specification with *Data* replaced by 1 . . 12. That is, every behavior allowed by your specification satisfies the specification Spec of module *Channel*, with *Data* replaced by the set 1 . . 12. Use *TLC* to check that Version 2 implements the specification *HourClockChannel* that you wrote in Exercise 1 of folder *AsynchronousInterface*.

   (c) Write specifications that hides the clock in the specifications of part (a). Explain informally why the resulting specification is equivalent to:

   - The *Channel* specification with *Data* replaced by 1 . . 12, for Version 1.

   - The *HourClockChannel* specification, for Version 2.

EXTENDS *Naturals*, *HourClock*, *Channel*, *TLC*

$$PrintVal(id,\ exp) \triangleq Print(\langle id,\ exp\rangle,\ \text{TRUE})$$

$$IsSending \triangleq chan.rdy \neq chan.ack$$

$$\begin{aligned} HCChannelInit \triangleq\ & \land\ TypeInvariant \\ & \land\ chan.ack = chan.rdy \\ & \land\ HCini \end{aligned}$$

$$\begin{aligned} HCChannelSend(d) \triangleq\ & \land\ chan.rdy = chan.ack \\ & \land\ chan' = [chan\ \text{EXCEPT}\ !.val = d,\ !.rdy = 1 - @] \\ & \land\ \text{UNCHANGED}\ \langle hr\rangle \end{aligned}$$

$$\begin{aligned} HCChannelRcv \triangleq\ & \land\ chan.rdy \neq chan.ack \\ & \land\ chan' = [chan\ \text{EXCEPT}\ !.ack = 1 - @] \\ & \land\ \text{UNCHANGED}\ \langle hr\rangle \end{aligned}$$

$$\begin{aligned} HCnxtChannel \triangleq\ & \land\ HCnxt \\ & \land\ \neg IsSending \\ & \land\ \text{UNCHANGED}\ \langle chan\rangle \\ & \land\ PrintVal(\text{“HCnxtChannel”},\ \langle hr,\ chan\rangle) \end{aligned}$$

$$HCChannelNext \triangleq (\exists\, d \in Data : HCChannelSend(d)) \lor HCChannelRcv \lor HCnxtChannel$$

$$HCChannelSpec \triangleq HCChannelInit \land \Box[HCChannelNext]_{\langle chan,\ hr\rangle}$$

THEOREM $HCChannelSpec \Rightarrow \Box TypeInvariant$

* Modification History
* Last modified Sat May 18 10:30:14 *PDT* 2019 by *jasondebolt*
* Created Sat May 18 09:31:23 *PDT* 2019 by *jasondebolt*