Checking Implementation

Open module *PCalBoundedBuffer* in the Toolbox and create a small model that substitutes 4 for N and a set of three model values for *Msg*. (For example set *Msg* to $\{m1, m2, m\}$ and choose the Set of model values option.) Add the formula $C!Spec$ to the Properties list in the What to check? section of the Model Overview page of the model, and run TLC. It should find no error.

Now, let's introduce an error. In the Definition Override section of the model's Advanced Options page, override the definition of *chBar* with the following definition.

$[i \in 1 \,..\, (p \ominus c) \mapsto$	`[i \in 1..(p (-) c) |->`
$\quad$ IF $p \ominus c = N$ THEN $buf[0]$	`     IF p (-) c = N THEN buf[0]`
$\qquad\qquad\qquad\quad$ ELSE $buf[(c + i - 1)\%N]]$	`                    ELSE buf[(c + i - 1) % N]]`

This changes the definition of *chBar* when $p \ominus c$ equals N, so it should introduce an error when the length of the sequence of sent messages reaches N, which can occur only after at least N steps.

Running TLC should now produce an error. Clicking on the location of the error leads to the formula $\Box[Next]_{vars}$ in module *PCalBoundedChannel*, indicating that the bounded buffer specification does not satisfy the property $\overline{\Box[Next]_{vars}}$. (Here and in the rest of this pop-up, *Next* is the formula by that name in module *PCalBoundedChannel*.)

To see why that property is violated, use the trace explorer to display the values of *chBar* during the execution. In the Error-Trace Exploration section of the TLC Errors window, use the Add button to enter the expression *chBar*. Click on the Explore button to run the trace explorer. The behavior shown in the Error-Trace section should now show the value of *chBar* in each state. Let's call that behavior σ. The behavior $\overline{\sigma}$ is defined to be the one whose i^{th} state assigns to the variable *ch* the value of *chBar* in the i^{th} state of σ. The formula $\overline{\Box[Next]_{vars}}$ is true of σ iff $\Box[Next]_{vars}$ is true of $\overline{\sigma}$, and $\Box[Next]_{vars}$ is not true of $\overline{\sigma}$ because the last step of $\overline{\sigma}$ does not satisfy $[Next]_{vars}$.

CLOSE