─────────────────────── MODULE *MinMax2* ───────────────────────

This module specifies a system with the same interaction between a user and a server as the one in module *MinMax1*, but instead of remembering the entire set of inputs, it uses two variables *min* and *max* to keep the largest and smallest values input thus far. Initially *min* equals Infinity and *max* equals *MinusInfinity*, where *Infinity* and *MinusInfinity* are two values that are considered greater than and less than any integer, respectively.

EXTENDS *Integers*, *Sequences*

CONSTANTS *Lo*, *Hi*, *Both*, *None*
ASSUME $\{Lo, Hi, Both, None\} \cap Int = \{\}$

$Infinity \triangleq$ CHOOSE $n : n \notin Int$
$MinusInfinity \triangleq$ CHOOSE $n : n \notin (Int \cup \{Infinity\})$

The operators *IsLeq* and *IsGeq* extend $\leq$ and $\geq$, respectively, to have the correct meaning when *Infinity* or *MinusInfinity* is one of the arguments.

$IsLeq(i, j) \triangleq (j = Infinity) \vee (i \leq j)$
$IsGeq(i, j) \triangleq (j = MinusInfinity) \vee (i \geq j)$

The rest of the specification is straightforward.

VARIABLES *x*, *turn*, *min*, *max*
$vars \triangleq \langle x, turn, min, max \rangle$

$Init \triangleq \quad \wedge x = None$
$\qquad\qquad \wedge turn = \text{``input''}$
$\qquad\qquad \wedge min = Infinity$
$\qquad\qquad \wedge max = MinusInfinity$

$InputNum \triangleq \quad \wedge turn = \text{``input''}$
$\qquad\qquad\qquad \wedge turn' = \text{``output''}$
$\qquad\qquad\qquad \wedge x' \in Int$
$\qquad\qquad\qquad \wedge$ UNCHANGED $\langle min, max \rangle$

$Respond \triangleq \quad \wedge turn = \text{``output''}$
$\qquad\qquad\qquad \wedge turn' = \text{``input''}$
$\qquad\qquad\qquad \wedge min' =$ IF $IsLeq(x, min)$ THEN x ELSE min
$\qquad\qquad\qquad \wedge max' =$ IF $IsGeq(x, max)$ THEN x ELSE max
$\qquad\qquad\qquad \wedge x' =$ IF $x = max'$ THEN IF $x = min'$ THEN $Both$ ELSE Hi
$\qquad\qquad\qquad\qquad\qquad\qquad$ ELSE IF $x = min'$ THEN Lo ELSE $None$

$Next \triangleq InputNum \vee Respond$

$Spec \triangleq Init \wedge \square[Next]_{vars}$

───

* Modification History
* Last modified *Fri Oct* 21 23:57:32 *PDT* 2016 by *lamport*
* Created *Fri Aug* 26 14:32:23 *PDT* 2016 by *lamport*

1