

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

$Infinity \triangleq \text{CHOOSE } n : n \notin Int$

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

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 $\triangleq \langle x, turn, min, max \rangle$

$Init \triangleq \wedge x = None$

$\wedge turn = \text{"input"}$

$\wedge min = Infinity$

$\wedge max = MinusInfinity$

$InputNum \triangleq \wedge turn = \text{"input"}$

$\wedge turn' = \text{"output"}$

$\wedge x' \in Int$

$\wedge \text{UNCHANGED } \langle min, max \rangle$

$Respond \triangleq \wedge turn = \text{"output"}$

$\wedge turn' = \text{"input"}$

$\wedge min' = \text{IF } IsLeq(x, min) \text{ THEN } x \text{ ELSE } min$

$\wedge max' = \text{IF } IsGeq(x, max) \text{ THEN } x \text{ ELSE } max$

$\wedge x' = \text{IF } x = max' \text{ THEN IF } x = min' \text{ THEN } Both \text{ ELSE } Hi$

$\text{ELSE IF } x = min' \text{ THEN } Lo \text{ ELSE } None$

The next-state relation and the spec.

$Next \triangleq InputNum \vee Respond$

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

The property to check (added by hengxin).

$NoneProperty \triangleq \Box [\wedge x \in Int$

$\wedge \{min, max\} \neq \{Infinity, MinusInfinity\}]$

```
\* Modification History
\* Last modified Tue Jun 26 20:48:02 CST 2018 by hengxin
\* Last modified Tue Jun 26 20:02:15 CST 2018 by hengxin
\* Last modified Fri Oct 21 23:57:32 PDT 2016 by lamport
\* Created Fri Aug 26 14:32:23 PDT 2016 by lamport
```