MODULE *SendInt1P*

This module is part of the example from the paper "Auxiliary Variables in TLA+" that also includes module *SendInt*1 and *SendInt*2. It adds a one-prediction prophecy variable p to specification *Spec* of *SendInt*1 to obtain specification *SpecP*, and it defines a refinement mapping under which *SpecP* implements specification *Spec* of module *SendInt*2.

EXTENDS *SendInt*1

Pi is the set of possible values of (predictions made by) p.

$Pi \triangleq Int$

The operator *PredSend* is used in the definition of *SendP* below. We define it before the declaration of the variable p to allow us more easily to check the theorem that follows it. This theorem asserts condition (4.9) of "Auxiliary Variables in TLA+", which ensures that $\exists\, p : SpecP$ (the specification obtained by hiding p in *SpecP*) is equivalent to *Spec*. To check this theorem with *TLC*, temporarily end the module by adding few " = " characters after the theorem and create a model having *Spec* as the specification.

$PredSend(i) \triangleq x' = i$

THEOREM $Spec \Rightarrow \Box[Send \Rightarrow \exists\, i \in Pi : PredSend(i)]_x$

VARIABLE p

$varsP \triangleq \langle x, p \rangle$

$InitP \triangleq Init \land (p \in Pi)$

$TypeOKP \triangleq TypeOK \land (p \in Pi)$

$SendP \triangleq Send \land PredSend(p) \land (p' \in Pi)$

$RcvP \triangleq Rcv \land (p' = p)$

$NextP \triangleq SendP \lor RcvP$

$SpecP \triangleq InitP \land \Box[NextP]_{varsP}$

The theorem below asserts that *SpecP* implements *SendInt*2 under the refinement mapping defined by the INSTANCE statement.

$SI2 \triangleq$ INSTANCE *SendInt*2 WITH $z \leftarrow$ IF $x = NotInt$ THEN p ELSE $NotInt$

THEOREM $SpecP \Rightarrow SI2!Spec$

* Modification History
* Last modified Sat *Oct* 22 00:37:21 *PDT* 2016 by *lamport*
* Created *Tue Sep* 06 02:21:32 *PDT* 2016 by *lamport*