MODULE *SendSetUndoP*

This module is part of the example in Section 4.4 of the paper "Auxiliary Variables in TLA+". It defines the specification *SpecUP* obtained, as explained in that paper, by adding a prophecy-array variable p to specification *SpecU* of module *SendSetUndo*. It then defines a refinement mapping under which *SpecUP* implements specification *Spec* of module *SendSet* to show that *SpecUP* implements $\exists\, y : Spec$.

EXTENDS *SendSetUndo*

The value of the variable p is always a function with domain y, which we call *Dom* to conform to the notation of Section 4.4 of "Auxiliary Variables in TLA+". The value of $p[d]$ predicts whether the element d of y will be sent or simply undone–that is, removed from y by an $Undo(S)$ action. These two predictions are represented by the string values "send" and "undo".

$$Pi \triangleq \{\text{"send"}, \text{"undo"}\}$$
$$Dom \triangleq y$$

As explained in the paper, the *SpecUP* is obtained by replacing each subaction A of *SpecU* with the subaction

$$Ap \triangleq \text{A} \land PredA(p) \land (p' \in NewPSetA(p))$$

For the reason explained below, we define *PredA* and *NewPSetA* before the variable p is declared, so we must define them to be operators that take p as an argument in the definition of Ap.

The definitions of *PredA* and *NewPSetA* are given below, for the four subactions A in the obvious disjunctive representation of *NextU*. (For example, *PredChoose* is *PredA* for A equal to *Choose*.)

$$PredChoose(p) \triangleq \text{TRUE}$$
$$NewPSetChoose(p) \triangleq \{f \in [Dom' \to Pi] \ : \ \forall\, d \in Dom : f[d] = p[d]\}$$

$$PredSend(p) \triangleq p[x'] = \text{"send"}$$
$$NewPSetSend(p) \triangleq \{[d \in Dom' \mapsto p[d]]\}$$

$$PredRcv(p) \triangleq \text{TRUE}$$
$$NewPSetRcv(p) \triangleq \{p\}$$

$$PredUndo(p, S) \triangleq \forall\, d \in S : p[d] = \text{"undo"}$$
$$NewPSetUndo(p) \triangleq \{[d \in Dom' \mapsto p[d]]\}$$

The following theorem must hold for p to be an auxiliary variable–that is, for $\exists\, p : SpecUP$ to be equivalent to *SpecU*. It is equivalent to the conjunction of (4.11) of "Auxiliary Variables in TLA+" for the four subactions A. Note that we need each *PredA* to be an operator in order to state this condition. We make *NewPSetA* an operator as well so we can put the two definitions together.

To check this theorem with TLA+, the module must be temporarily ended after the theorem so a model can be created having *SpecU* as its specification.

$$\begin{aligned} Condition \triangleq \ \Box[& \land Choose \Rightarrow \exists f \in [Dom \to Pi] : PredChoose(f) \\ & \land Send \ \Rightarrow \exists f \in [Dom \to Pi] : PredSend(f) \\ & \land Rcv \ \Rightarrow \exists f \in [Dom \to Pi] : PredRcv(f) \\ & \land \forall\, S \in (\text{SUBSET } y) \setminus \{\{\}\} \ : \\ & \qquad Undo(S) \Rightarrow \exists f \in [Dom \to Pi] : PredUndo(f, S) \\ ]_{vars} & \end{aligned}$$

THEOREM $SpecU \Rightarrow Condition$

VARIABLE p

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

$TypeOKP \triangleq TypeOK \land (p \in [Dom \to Pi])$

Since Dom equals y, which initially equals the empty set, the initial value of p is the unique function with empty domain.

$InitUP \triangleq Init \land (p = \langle\rangle)$

The rest of the specification is as explained above.

$$\begin{array}{lcl} ChooseP & \triangleq & Choose \land PredChoose(p) \land (p' \in NewPSetChoose(p)) \\ SendP & \triangleq & Send \land PredSend(p) \land (p' \in NewPSetSend(p)) \\ RcvP & \triangleq & Rcv \land PredRcv(p) \land (p' \in NewPSetRcv(p)) \\ UndoP(S) & \triangleq & Undo(S) \land PredUndo(p,\ S) \land (p' \in NewPSetUndo(p)) \end{array}$$

$NextUP \triangleq ChooseP \lor SendP \lor RcvP \lor (\exists\, S \in (\text{SUBSET } y) \setminus \{\{\}\} : UndoP(S))$

$SpecUP \triangleq InitUP \land \Box[NextUP]_{varsP}$

The INSTANCE statement and theorem assert that *SpecUP* implements specification *Spec* of module *SendSet* under the indicated refinement mapping.

$SS \triangleq \text{INSTANCE } SendSet \text{ WITH } y \leftarrow \{d \in y : p[d] = \text{“send”}\}$

THEOREM $SpecUP \Rightarrow SS\,!\,Spec$

* Modification History
* Last modified Sat *Oct* 22 00:47:46 *PDT* 2016 by *lamport*
* Created Sun *Sep* 25 05:58:07 *PDT* 2016 by *lamport*