

The first part of this module defines the TLA+ representation of the abstract program *ICenSet1* of Figure 7.8 of the book “A Science of Computer Programs” by *Leslie Lamport*. The second part defines a program *ICen1Set_p* obtained by adding a prophecy set variable *p* to *ICenSet1* as described in Section 7.4.3.2, and it asserts that *ICen1Set_p* implements *ICenSet2* under a suitable refinement mapping. For that purpose, it instantiates module *ICenSet2* that defines program *ICenSet2*.

EXTENDS *Integers, Sequences*

CONSTANT *Art, NotArt*

VARIABLES *inp, aw, disp, old*

$v \triangleq \langle inp, aw, disp, old \rangle$

$TypeOKSet \triangleq \wedge inp \in Art \cup \{NotArt\}$
 $\wedge disp \in Art \times \{0, 1\}$
 $\wedge aw \subseteq Art$
 $\wedge old \subseteq Art$

$InitSet \triangleq \wedge inp = NotArt$
 $\wedge aw = \{\}$
 $\wedge disp \in Art \times \{0, 1\}$
 $\wedge old = \{\}$

$InputSet \triangleq \wedge inp = NotArt$
 $\wedge inp' \in Art \setminus old$
 $\wedge aw' = aw \cup \{inp'\}$
 $\wedge old' = old \cup \{inp'\}$
 $\wedge disp' = disp$

$AckSet \triangleq \wedge inp \in Art$
 $\wedge inp' = NotArt$
 $\wedge UNCHANGED \langle aw, disp, old \rangle$

$DispOrNotSet \triangleq \exists w \in aw :$
 $\wedge \vee disp' = \langle w, 1 - disp[2] \rangle$
 $\vee disp' = disp$
 $\wedge aw' = aw \setminus \{w\}$
 $\wedge UNCHANGED \langle inp, old \rangle$

$NextSet1 \triangleq InputSet \vee AckSet \vee DispOrNotSet$

$ICenSet1 \triangleq InitSet \wedge \Box[NextSet1]_v$

Although not done in the book, we define a fairness requirement for the *ICenSet1* program. It is weak fairness of all the actions except the *InputSet* action, since we don't want to require that the artist keep submitting works of art.

$Fairness1 \triangleq WF_v(AckSet \vee DispOrNotSet)$

$$ICensSet1Live \triangleq ICenSet1 \wedge Fairness1$$

We now define the program $ICenSet1_p$ obtained by adding a variable p that makes a set of predictions, as described in Section 7.4.3.2.

CONSTANTS Yes, No

$Pi \triangleq \{Yes, No\}$

ASSUME $\wedge Yes \neq No$
 $\wedge Pi \cap Art = \{\}$

$FcnPlus(f, w, d) \triangleq$

$[x \in \{w\} \cup \text{DOMAIN } f \mapsto \text{IF } x = w \text{ THEN } d \text{ ELSE } f[x]]$

$FcnMinus(f, w) \triangleq [x \in (\text{DOMAIN } f) \setminus \{w\} \mapsto f[x]]$

VARIABLE p

$v_p \triangleq \langle inp, aw, disp, old, p \rangle$

$TypeOKSet_p \triangleq TypeOKSet \wedge (p \in [aw \rightarrow Pi])$

$InitSet_p \triangleq InitSet \wedge (p = \langle \rangle)$

$InputSet_p \triangleq InputSet \wedge (\exists i \in Pi : p' = FcnPlus(p, inp', i))$

$AckSet_p \triangleq AckSet \wedge (p' = p)$

$DorNSet(w, i) \triangleq$
 $\wedge \vee (i = Yes) \wedge (disp' = \langle w, 1 - disp[2] \rangle)$
 $\vee (i = No) \wedge (disp' = disp)$
 $\wedge aw' = aw \setminus \{w\}$
 $\wedge \text{UNCHANGED } \langle inp, old \rangle$

$DispOrNotSet_p \triangleq \exists w \in aw :$
 $\wedge DorNSet(w, p[w])$
 $\wedge p' = FcnMinus(p, w)$

$NextSet1_p \triangleq InputSet_p \vee AckSet_p \vee DispOrNotSet_p$

$ICenSet1_p \triangleq InitSet_p \wedge \Box[NextSet1_p]_{v_p} \wedge Fairness1$

The program $ICenSet2$ is defined in module $ICenSet2$. We now define $awBar$ as in the book and instantiate that module to define $IC2!ICenSet2$ to be $ICenSet2$ WITH $aw \leftarrow awBar$, and we assert that it is implied by $ICenSet1_p$.

$awBar \triangleq \{w \in aw : p[w] = Yes\}$

$IC2 \triangleq \text{INSTANCE } ICenSet2 \text{ WITH } aw \leftarrow awBar$

THEOREM $ICenSet1 \Rightarrow IC2!ICenSet2$

\ * Modification History

\ * Last modified *Fri Oct 18 15:16:28 CEST 2024* by *lamport*

\ * Created *Sat Oct 22 07:58:56 PDT 2022* by *lamport*