MODULE *AfekSimplifiedH*

This module adds a history variable h to the algorithm in module *AfekSimplified* and shows that the resulting specification *SpecH* implements specification *SafeSpec* of module *NewLinearSnapshot* under a suitable refinement mapping. This shows that specification *Spec* of module *AfekSimplified* implies $\exists\, mem, rstate, wstate : SafeSpec$.

The history variable h is modified by the *BeginRd*, *EndRd*, and *DoWr* actions exactly the way the corresponding actions of *NewLinearSnapshot* change *rstate*, so the refinement mapping can instantiate *rstate* with h. The instantiations of the other internal variables of *NewLinearSnapshot* are straightforward.

EXTENDS *AfekSimplified*, *Sequences*

VARIABLE h

$varsH \triangleq \langle vars, h\rangle$

$TypeOKH \triangleq TypeOK \land (h \in [Readers \to Seq(MemVals)])$

$InitH \triangleq Init \land (h = [i \in Readers \mapsto \langle\rangle])$

We define *memBar* to be the value of the variable *mem* of *NewLinearSnapshot* represented by *imem*. It is used both to instantiate the variable *mem* and in the definitions of the value of h' in some of the actions.

$memBar \triangleq [i \in Writers \mapsto imem[i][1]]$

$BeginWrH(i, cmd) \triangleq BeginWr(i, cmd) \land (h' = h)$

$$\begin{aligned} DoWrH(i) \triangleq\ & \land DoWr(i) \\ & \land h' = [j \in Readers \mapsto \\ & \qquad \text{IF } h[j] = \langle\rangle \\ & \qquad\ \text{THEN } \langle\rangle \\ & \qquad\ \text{ELSE } Append(h[j], memBar')] \end{aligned}$$

$EndWrH(i) \triangleq EndWr(i) \land (h' = h)$

$$\begin{aligned} BeginRdH(i) \triangleq\ & \land BeginRd(i) \\ & \land h' = [h \text{ EXCEPT } ![i] = \langle memBar\rangle] \end{aligned}$$

$Rd1H(i) \triangleq Rd1(i) \land (h' = h)$

$Rd2H(i) \triangleq Rd2(i) \land (h' = h)$

$$\begin{aligned} TryEndRdH(i) \triangleq\ & \land TryEndRd(i) \\ & \land h' = \text{IF } rdVal1[i] = rdVal2[i] \\ & \qquad\ \text{THEN } [h \text{ EXCEPT } ![i] = \langle\rangle] \\ & \qquad\ \text{ELSE } h \end{aligned}$$

$$\begin{aligned} NextH \triangleq\ & \\ & \lor \exists\, i \in Readers : BeginRdH(i) \lor Rd1H(i) \lor Rd2H(i) \lor TryEndRdH(i) \\ & \lor \exists\, i \in Writers : \lor \exists\, cmd \in RegVals : BeginWrH(i, cmd) \\ & \qquad\qquad\ \lor DoWrH(i) \lor EndWrH(i) \end{aligned}$$

$SpecH \triangleq InitH \wedge \Box[NextH]_{varsH}$

We instantiate *wstate* with the following expression *wstateBar*.

$$
\begin{array}{l}
wstateBar \triangleq [i \in Writers \mapsto \\
\quad \text{IF } (interface[i] = NotRegVal) \vee (wrNum[i] = imem[i][2]) \\
\quad \text{THEN } NotRegVal \\
\quad \text{ELSE } interface[i]]
\end{array}
$$

Here is the INSTANCE statement and theorem asserting that *SpecH* implements *SafeSpec* of module *NewLinearSnapshot* under the refinement mapping. This theorem implies that the algorithm implements

$\exists\, mem, rstate, wstate : Spec$

where *Spec* is the specification in *NewLinearSnapshot*.

$$
\begin{array}{l}
NLS \triangleq \text{INSTANCE } NewLinearSnapshot \\
\qquad \text{WITH } mem \leftarrow memBar,\ rstate \leftarrow h,\ wstate \leftarrow wstateBar
\end{array}
$$

THEOREM $SpecH \Rightarrow NLS!SafeSpec$

* Modification History
* Last modified Sat *Oct* 22 02:03:17 *PDT* 2016 by *lamport*
* Created *Wed Oct* 05 09:45:14 *PDT* 2016 by *lamport*