MODULE *SSIRefinement*

EXTENDS *SSI*, *Sequences*, *FiniteSets*

CONSTANTS *NULL*, *Flip*, *Flop*

VARIABLES *h*, *fateIsSet*, *canIssue*, *parity*, *reads*, *writes*, *ord*, *tenvBar*

Refinement transactions

$TrR \triangleq Tr \setminus \{T0\}$

Committed transactions

$CT \triangleq \{t \in TrR : tstate[t] = Committed\}$

$N \triangleq Cardinality(CT)$

$$
\begin{array}{ll}
TypeOkR \triangleq & \land\ TypeOkS \\
& \land\ \forall\, i \in \text{DOMAIN } h : \text{LET } e \triangleq h[i] \text{ IN} \\
& \quad \land\ e.tr \in TrR \\
& \quad \land\ e.op \in \{\text{“r”}, \text{“w”}, \text{“c”}, \text{“a”}\} \\
& \quad \land\ e.arg \in \text{CASE } e.op = \text{“r”} \to Obj \\
& \qquad\qquad\qquad\ \square\ e.op = \text{“w”} \to Obj \times Val \\
& \qquad\qquad\qquad\ \square\ \text{OTHER} \to \{\langle\rangle\} \\
& \quad \land\ e.rval \in Val \cup \{Ok, Err\} \\
& \quad \land\ e.tstate \in [Tr \to \{Unstarted, Open, Committed, Aborted\}] \\
& \quad \land\ e.op \in \{\text{“r”}, \text{“w”}\} \Rightarrow\ \land\ \text{DOMAIN } e.wr \subseteq Obj \\
& \qquad\qquad\qquad\qquad\qquad\qquad\ \land\ \forall\, obj \in \text{DOMAIN } e.wr : e.wr[obj] \in Val \\
& \land\ fateIsSet \in \text{BOOLEAN} \\
& \land\ canIssue \in \text{BOOLEAN} \\
& \land\ parity \in \{0, 1\} \\
& \land\ reads \in [Tr \to \text{SUBSET } Obj] \\
& \land\ writes \in [Tr \to \text{SUBSET } Obj] \\
& \land\ tenvBar \in [CT \to [Obj \to Val]] \cup \{NULL\} \\
& \land\ ord \in [to : [1 \,..\, N \to CT] \cup \{NULL\},\ benv : [1 \,..\, N+1 \to [Obj \to Val]] \cup \{NULL\}]
\end{array}
$$

$$
\begin{array}{ll}
InitR \triangleq & \land\ InitS \\
& \land\ fateIsSet = \text{FALSE} \\
& \land\ parity = 0 \\
& \land\ h = \langle\rangle \\
& \land\ canIssue = \text{FALSE} \\
& \land\ reads = [t \in Tr \mapsto \{\}] \\
& \land\ writes = [t \in Tr \mapsto \text{IF } t = T0 \text{ THEN } Obj \text{ ELSE } \{\}] \\
& \land\ ord = [to \mapsto NULL,\ benv \mapsto NULL] \\
& \land\ tenvBar = NULL
\end{array}
$$

$$
\begin{array}{ll}
StartTransactionR(t) \triangleq & \land\ StartTransactionS(t) \\
& \land\ \text{UNCHANGED } \langle h, fateIsSet, canIssue, parity, reads, writes, ord, tenvBar\rangle
\end{array}
$$

$$BeginRdR(t,\ obj) \triangleq \land\ BeginRdS(t,\ obj)$$
$$\land\ \text{UNCHANGED}\ \langle h,\ fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$$

$$AbortRdR(t,\ obj) \triangleq$$
$$\land\ AbortRdS(t,\ obj)$$
$$\land\ h' = Append(h,\ [tr \mapsto t,\ op \mapsto \text{“a”},\ arg \mapsto \langle\rangle,\ rval \mapsto Err,$$
$$tstate \mapsto [tstate\ \text{EXCEPT}\ ![t] = Aborted]])$$
$$\land\ \text{UNCHANGED}\ \langle fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$$

$$EndRdR(t,\ obj,\ val) \triangleq$$
$$\land\ EndRdS(t,\ obj,\ val)$$
$$\land\ h' = Append(h,\ [tr \mapsto t,\ op \mapsto \text{“r”},\ arg \mapsto obj,\ rval \mapsto val,$$
$$tstate \mapsto tstate,\ wr \mapsto [o \in writes[t] \mapsto Get(t,\ o)]])$$
$$\land\ reads' = \text{IF}\ obj \in writes[t]\ \text{THEN}\ reads\ \text{ELSE}\ [reads\ \text{EXCEPT}\ ![t] = @ \cup \{obj\}]\ \text{unwritten reads}$$
$$\land\ parity' = 1 - parity$$
$$\land\ \text{UNCHANGED}\ \langle fateIsSet,\ canIssue,\ writes,\ ord,\ tenvBar\rangle$$

$$BeginWrR(t,\ obj,\ val) \triangleq \land\ BeginWrS(t,\ obj,\ val)$$
$$\land\ \text{UNCHANGED}\ \langle h,\ fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$$

$$EndWrR(t,\ obj,\ val) \triangleq$$
$$\land\ EndWrS(t,\ obj,\ val)$$
$$\land\ h' = Append(h,\ [tr \mapsto t,\ op \mapsto \text{“w”},\ arg \mapsto \langle obj,\ val\rangle,\ rval \mapsto Ok,$$
$$tstate \mapsto tstate,\ wr \mapsto [o \in writes[t] \mapsto Get(t,\ o)]])$$
$$\land\ writes' = [writes\ \text{EXCEPT}\ ![t] = @ \cup \{obj\}]$$
$$\land\ parity' = 1 - parity$$
$$\land\ \text{UNCHANGED}\ \langle fateIsSet,\ canIssue,\ reads,\ ord,\ tenvBar\rangle$$

$$AbortWrR(t,\ obj) \triangleq$$
$$\land\ AbortWrS(t,\ obj)$$
$$\land\ h' = Append(h,\ [tr \mapsto t,\ op \mapsto \text{“a”},\ arg \mapsto \langle\rangle,\ rval \mapsto Err,$$
$$tstate \mapsto [tstate\ \text{EXCEPT}\ ![t] = Aborted]])$$
$$\land\ \text{UNCHANGED}\ \langle fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$$

$$BeginCommitR(t) \triangleq \land\ BeginCommit(t)$$
$$\land\ \text{UNCHANGED}\ \langle h,\ fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$$

$$AbortCommitR(t) \triangleq$$
$$\land\ AbortCommit(t)$$
$$\land\ h' = Append(h,\ [tr \mapsto t,\ op \mapsto \text{“a”},\ arg \mapsto \langle\rangle,\ rval \mapsto Ok,$$
$$tstate \mapsto [tstate\ \text{EXCEPT}\ ![t] = Aborted]])$$
$$\land\ \text{UNCHANGED}\ \langle fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$$

$$EndCommitR(t) \triangleq$$
$$\land\ EndCommit(t)$$
$$\land\ h' = Append(h,\ [tr \mapsto t,\ op \mapsto \text{“c”},\ arg \mapsto \langle\rangle,\ rval \mapsto Ok,$$
$$tstate \mapsto [tstate\ \text{EXCEPT}\ ![t] = Committed]])$$

$\land$ UNCHANGED $\langle fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$

$AbortR(t) \triangleq$
$\land\ AbortS(t)$
$\land\ h' = Append(h,\ [tr \mapsto t,\ op \mapsto$ "a", $arg \mapsto \langle\rangle,\ rval \mapsto Ok,$
$tstate \mapsto [tstate$ EXCEPT $![t] = Aborted]])$
$\land$ UNCHANGED $\langle fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$

$DetectDeadlockR \triangleq \land\ DetectDeadlockS$
$\land$ UNCHANGED $\langle h,\ fateIsSet,\ canIssue,\ parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$

Get the order in which this transactionruns
$Ord(t) \triangleq$ CHOOSE $i \in$ DOMAIN $ord.to : ord.to[i] = t$

$SetFate \triangleq \land\ Done$
$\land\ fateIsSet =$ FALSE
$\land\ fateIsSet' =$ TRUE
$\land\ ord' =$ CHOOSE $r \in [to : [1\ ..\ N \to CT],\ benv : [1\ ..\ N+1 \to [Obj \to Val]]] :$
first environment must be the initialization
$\land\ r.benv[1] = SnapInit$
to must be a total ordering
$\land \forall\ i,\ j \in 1\ ..\ N : r.to[i] = r.to[j] \Rightarrow i = j$
$\land \forall\ i \in 1\ ..\ N :$ LET $t \triangleq r.to[i]$IN
all non-written reads have to be consistent with transaction's snapshot
$\land \forall\ obj \in reads[t] :\ r.benv[i][obj] = GetVer(obj,\ vis[t] \setminus \{t\}).val$
all writes have to be consistent with transaction's environment
$\land \forall\ obj \in writes[t] : r.benv[i+1][obj] = Get(t,\ obj)$
if a variable changed, there must be a corresponding write
$\land \forall\ obj \in Obj : (r.benv[i+1][obj] \neq r.benv[i][obj]) \Rightarrow obj \in writes[t]$
$\land\ tenvBar' =$ LET $ordp \triangleq ord'$
$benv \triangleq ordp.benv$
$to \triangleq ordp.to$IN
$[t \in CT \mapsto$ LET $i \triangleq$ CHOOSE $i \in$ DOMAIN $to : to[i] = t$IN $benv[i]]$
$\land$ UNCHANGED $\langle op,\ arg,\ rval,\ tr,\ db,\ vis,\ tstate,\ tid,\ deadlocked,$
$h,\ canIssue,\ parity,\ reads,\ writes,\ rds,\ inc,\ outc\rangle$

$Issue \triangleq \land\ h \neq \langle\rangle$
$\land\ fateIsSet$
$\land\ canIssue' =$ TRUE
$\land\ h' =$ IF $canIssue$ THEN $Tail(h)$ ELSE h
$\land\ h' \neq \langle\rangle$
$tenvBar'$ needs to reflect the state of the *next* head in the history, not the current head
$\land\ tenvBar' =$ LET $e \triangleq Head(h')$
$obj \triangleq e.arg[1]$
$val \triangleq e.arg[2]$
$t \triangleq e.tr$

$$
\begin{array}{l}
\quad\ \text{IN}\quad \text{IF } tstate[e.tr] = Committed \land e.op = \text{“w”} \\
\qquad\ \text{THEN } [tenvBar \text{ EXCEPT } ![t][obj] = val] \\
\qquad\ \text{ELSE } tenvBar \\
\land \text{UNCHANGED } \langle op,\ arg,\ rval,\ tr,\ db,\ vis,\ tstate,\ tid,\ deadlocked, \\
\qquad\qquad fateIsSet,\ parity,\ reads,\ writes,\ ord,\ rds,\ inc,\ outc\rangle
\end{array}
$$

$$
\begin{array}{l}
vv \triangleq \langle op,\ arg,\ rval,\ tr,\ db,\ vis,\ tstate,\ tid,\ deadlocked,\ h,\ fateIsSet,\ canIssue, \\
\qquad parity,\ reads,\ writes,\ ord,\ tenvBar,\ rds,\ inc,\ outc\rangle
\end{array}
$$

$$
\begin{array}{l}
TerminationR \triangleq \land\ Done \\
\qquad\qquad\qquad\ \land\ Tail(h) = \langle\rangle \\
\qquad\qquad\qquad\ \land \text{UNCHANGED } vv
\end{array}
$$

$$
\begin{array}{l}
NextR \triangleq \lor \exists\, t \in Tr,\ obj \in Obj,\ val \in Val : \\
\qquad\qquad \lor\ StartTransactionR(t) \\
\qquad\qquad \lor\ BeginRdR(t,\ obj) \\
\qquad\qquad \lor\ EndRdR(t,\ obj,\ val) \\
\qquad\qquad \lor\ AbortRdR(t,\ obj) \\
\qquad\qquad \lor\ BeginWrR(t,\ obj,\ val) \\
\qquad\qquad \lor\ EndWrR(t,\ obj,\ val) \\
\qquad\qquad \lor\ AbortWrR(t,\ obj) \\
\qquad\qquad \lor\ BeginCommitR(t) \\
\qquad\qquad \lor\ AbortCommitR(t) \\
\qquad\qquad \lor\ EndCommitR(t) \\
\qquad\qquad \lor\ AbortR(t) \\
\qquad\quad \lor\ DetectDeadlockR \\
\qquad\quad \lor\ Issue \\
\qquad\quad \lor\ SetFate \\
\qquad\quad \lor\ TerminationR
\end{array}
$$

$$
SpecR \triangleq InitR \land \Box[NextR]_{vv}
$$

$$
\begin{array}{l}
trBar \triangleq \text{IF } canIssue \text{ THEN } Head(h).tr \text{ ELSE } T0 \\
opBar \triangleq \text{IF } canIssue \text{ THEN } Head(h).op \text{ ELSE “r”} \\
argBar \triangleq \text{CASE } canIssue \land Head(h).arg = \langle\rangle \to None \\
\qquad\qquad\ \Box\ \ canIssue \land Head(h).arg \neq \langle\rangle \to Head(h).arg \\
\qquad\qquad\ \Box\ \ \text{OTHER } \to \text{CHOOSE } obj \in Obj : \text{TRUE} \\
rvalBar \triangleq \text{CASE } canIssue \land Head(h).rval \neq Err \to Head(h).rval \\
\qquad\qquad\ \Box\ \ canIssue \land Head(h).rval = Err \to Ok \\
\qquad\qquad\ \Box\ \ \text{OTHER} \qquad\qquad\qquad\qquad\ \to V0 \\
tstateBar \triangleq [t \in TrR \mapsto \\
\qquad\qquad \text{LET } s \triangleq Head(h).tstate[t] \text{IN} \\
\qquad\qquad \text{CASE } \neg canIssue \qquad\qquad\ \to Open \\
\qquad\qquad\quad \Box\ \ canIssue \land s = Unstarted \ \to Open \\
\qquad\qquad\quad \Box\ \ canIssue \land s = Open \qquad\ \to Open \\
\qquad\qquad\quad \Box\ \ canIssue \land s = Committed \to Committed
\end{array}
$$

$$\Box \quad canIssue \wedge s = Aborted \quad \rightarrow Aborted]$$

$ffBar \triangleq$ LET $Parity(hh) \triangleq Len(SelectSeq(hh,$ LAMBDA $e : e.op \in \{$"r", "w"$\}))\%2$
$p \triangleq Parity(h)$
$opp \triangleq Head(h).op$IN
CASE $\neg canIssue \rightarrow Flip$
$\Box \quad canIssue \wedge opp \in \{$"r", "w"$\} \wedge parity = p \rightarrow Flop$
$\Box \quad canIssue \wedge opp \notin \{$"r", "w"$\} \wedge parity = p \rightarrow Flip$
$\Box \quad canIssue \wedge opp \in \{$"r", "w"$\} \wedge parity \neq p \rightarrow Flip$
$\Box \quad canIssue \wedge opp \notin \{$"r", "w"$\} \wedge parity \neq p \rightarrow Flop$

$fateBar \triangleq$ IF $\neg fateIsSet$ THEN $NULL$
ELSE $[t \in TrR \mapsto tstate[t]]$

$Ser \triangleq$ INSTANCE $SerializabilityD$ WITH
$Tr \leftarrow TrR$,
$tr \leftarrow trBar$,
$op \leftarrow opBar$,
$arg \leftarrow argBar$,
$rval \leftarrow rvalBar$,
$tstate \leftarrow tstateBar$,
$fate \leftarrow fateBar$,
$to \leftarrow ord.to$,
$tenv \leftarrow tenvBar$,
$benv \leftarrow ord.benv$,
$ff \leftarrow ffBar$,
$Vinit \leftarrow V0$

$SerSpec \triangleq Ser!SpecD$