
MODULE *MVCCRefinement*

EXTENDS *MVCC*, *Naturals*, *Sequences*, *FiniteSets*, *TLC*

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

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

Refinement transactions

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

Committed transactions

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

$N \triangleq Cardinality(CT)$

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

$InitR \triangleq \wedge Init$
 $\wedge fateIsSet = FALSE$
 $\wedge parity = 0$
 $\wedge h = \langle \rangle$
 $\wedge canIssue = FALSE$
 $\wedge reads = [t \in Tr \mapsto \{\}]$
 $\wedge writes = [t \in Tr \mapsto IF\ t = T0\ THEN\ Obj\ ELSE\ \{\}]$
 $\wedge ord = [to \mapsto NULL, benv \mapsto NULL]$
 $\wedge tenvBar = NULL$

$StartTransactionR(t) \triangleq \wedge StartTransaction(t)$
 $\wedge UNCHANGED\ \langle h, fateIsSet, canIssue, parity, reads, writes, ord, tenvBar \rangle$

$$\begin{aligned}
& \wedge \text{fateIsSet}' = \text{TRUE} \\
& \wedge \text{ord}' = \text{CHOOSE } r \in [to : [1 \dots N \rightarrow CT], \text{benv} : [1 \dots N + 1 \rightarrow [Obj \rightarrow Val]]] : \\
& \quad \text{first environment must be the initialization} \\
& \quad \wedge r.\text{benv}[1] = \text{SnapInit} \\
& \quad \text{to must be a total ordering} \\
& \quad \wedge \forall i, j \in 1 \dots N : r.to[i] = r.to[j] \Rightarrow i = j \\
& \quad \wedge \forall i \in 1 \dots N : \text{LET } t \triangleq r.to[i] \text{ IN} \\
& \quad \quad \text{all non-written reads have to be consistent with transaction's snapshot} \\
& \quad \quad \wedge \forall obj \in reads[t] : r.\text{benv}[i][obj] = \text{GetVer}(obj, vis[t] \setminus \{t\}).val \\
& \quad \quad \text{all writes have to be consistent with transaction's environment} \\
& \quad \quad \wedge \forall obj \in writes[t] : r.\text{benv}[i + 1][obj] = \text{Get}(t, obj) \\
& \quad \quad \text{if a variable changed, there must be a corresponding write} \\
& \quad \quad \wedge \forall obj \in Obj : (r.\text{benv}[i + 1][obj] \neq r.\text{benv}[i][obj]) \Rightarrow obj \in writes[t] \\
& \wedge \text{tenvBar}' = \text{LET } \text{ordp} \triangleq \text{ord}' \\
& \quad \quad \text{benv} \triangleq \text{ordp}.\text{benv} \\
& \quad \quad to \triangleq \text{ordp}.to \text{ IN} \\
& \quad [t \in CT \mapsto \text{LET } i \triangleq \text{CHOOSE } i \in \text{DOMAIN } to : to[i] = t \text{ IN } \text{benv}[i]] \\
& \wedge \text{UNCHANGED } \langle op, arg, rval, tr, db, vis, tstate, tid, deadlocked, \\
& \quad h, canIssue, parity, reads, writes \rangle \\
\\
\text{Issue} & \triangleq \wedge h \neq \langle \rangle \\
& \wedge \text{fateIsSet} \\
& \wedge \text{canIssue}' = \text{TRUE} \\
& \wedge h' = \text{IF } \text{canIssue} \text{ THEN } \text{Tail}(h) \text{ ELSE } h \\
& \wedge h' \neq \langle \rangle \\
& \quad \text{tenvBar}' \text{ needs to reflect the state of the *next* head in the history, not the current head} \\
& \wedge \text{tenvBar}' = \text{LET } e \triangleq \text{Head}(h') \\
& \quad \quad obj \triangleq e.arg[1] \\
& \quad \quad val \triangleq e.arg[2] \\
& \quad \quad t \triangleq e.tr \\
& \quad \text{IN IF } tstate[e.tr] = \text{Committed} \wedge e.op = \text{"w"} \\
& \quad \quad \text{THEN } [tenvBar \text{ EXCEPT } ![t][obj] = val] \\
& \quad \quad \text{ELSE } tenvBar \\
& \wedge \text{UNCHANGED } \langle op, arg, rval, tr, db, vis, tstate, tid, deadlocked, \\
& \quad \text{fateIsSet, parity, reads, writes, ord} \rangle \\
\\
vv & \triangleq \langle op, arg, rval, tr, db, vis, tstate, tid, deadlocked, h, fateIsSet, canIssue, \\
& \quad parity, reads, writes, ord, tenvBar \rangle \\
\\
\text{TerminationR} & \triangleq \wedge \text{Done} \\
& \quad \wedge \text{Tail}(h) = \langle \rangle \\
& \quad \wedge \text{UNCHANGED } vv \\
\\
\text{NextR} & \triangleq \vee \exists t \in Tr, obj \in Obj, val \in Val : \\
& \quad \vee \text{StartTransactionR}(t)
\end{aligned}$$

$$\begin{aligned}
& \vee \text{BeginRdR}(t, \text{obj}) \\
& \vee \text{EndRdR}(t, \text{obj}, \text{val}) \\
& \vee \text{BeginWrR}(t, \text{obj}, \text{val}) \\
& \vee \text{EndWrR}(t, \text{obj}, \text{val}) \\
& \vee \text{AbortWrR}(t, \text{obj}, \text{val}) \\
& \vee \text{CommitR}(t) \\
& \vee \text{AbortR}(t) \\
& \vee \text{DetectDeadlockR} \\
& \vee \text{Issue} \\
& \vee \text{SetFate} \\
& \vee \text{TerminationR} \\
\text{SpecR} & \triangleq \text{InitR} \wedge \square[\text{NextR}]_{vv} \\
\text{trBar} & \triangleq \text{IF } \text{canIssue} \text{ THEN } \text{Head}(h).\text{tr} \text{ ELSE } T0 \\
\text{opBar} & \triangleq \text{IF } \text{canIssue} \text{ THEN } \text{Head}(h).\text{op} \text{ ELSE } \text{"r"} \\
\text{argBar} & \triangleq \text{CASE } \text{canIssue} \wedge \text{Head}(h).\text{arg} = \langle \rangle \rightarrow \text{None} \\
& \quad \square \text{canIssue} \wedge \text{Head}(h).\text{arg} \neq \langle \rangle \rightarrow \text{Head}(h).\text{arg} \\
& \quad \square \text{OTHER} \rightarrow \text{CHOOSE } \text{obj} \in \text{Obj} : \text{TRUE} \\
\text{rvalBar} & \triangleq \text{CASE } \text{canIssue} \wedge \text{Head}(h).\text{rval} \neq \text{Err} \rightarrow \text{Head}(h).\text{rval} \\
& \quad \square \text{canIssue} \wedge \text{Head}(h).\text{rval} = \text{Err} \rightarrow \text{Ok} \\
& \quad \square \text{OTHER} \rightarrow V0 \\
\text{tstateBar} & \triangleq [t \in \text{TrR} \mapsto \\
& \quad \text{LET } s \triangleq \text{Head}(h).\text{tstate}[t] \text{ IN} \\
& \quad \text{CASE } \neg \text{canIssue} \rightarrow \text{Open} \\
& \quad \square \text{canIssue} \wedge s = \text{Unstarted} \rightarrow \text{Open} \\
& \quad \square \text{canIssue} \wedge s = \text{Open} \rightarrow \text{Open} \\
& \quad \square \text{canIssue} \wedge s = \text{Committed} \rightarrow \text{Committed} \\
& \quad \square \text{canIssue} \wedge s = \text{Aborted} \rightarrow \text{Aborted}] \\
\text{ffBar} & \triangleq \text{LET } \text{Parity}(hh) \triangleq \text{Len}(\text{SelectSeq}(hh, \text{LAMBDA } e : e.\text{op} \in \{\text{"r"}, \text{"w"}\})) \% 2 \\
& \quad p \triangleq \text{Parity}(h) \\
& \quad \text{opp} \triangleq \text{Head}(h).\text{op} \text{ IN} \\
& \quad \text{CASE } \neg \text{canIssue} \rightarrow \text{Flip} \\
& \quad \square \text{canIssue} \wedge \text{opp} \in \{\text{"r"}, \text{"w"}\} \wedge \text{parity} = p \rightarrow \text{Flop} \\
& \quad \square \text{canIssue} \wedge \text{opp} \notin \{\text{"r"}, \text{"w"}\} \wedge \text{parity} = p \rightarrow \text{Flip} \\
& \quad \square \text{canIssue} \wedge \text{opp} \in \{\text{"r"}, \text{"w"}\} \wedge \text{parity} \neq p \rightarrow \text{Flip} \\
& \quad \square \text{canIssue} \wedge \text{opp} \notin \{\text{"r"}, \text{"w"}\} \wedge \text{parity} \neq p \rightarrow \text{Flop} \\
\text{fateBar} & \triangleq \text{IF } \neg \text{fateIsSet} \text{ THEN } \text{NULL} \\
& \quad \text{ELSE } [t \in \text{TrR} \mapsto \text{tstate}[t]] \\
\text{Ser} & \triangleq \text{INSTANCE } \text{SerializabilityD} \text{ WITH} \\
& \quad \text{Tr} \leftarrow \text{TrR}, \\
& \quad \text{tr} \leftarrow \text{trBar}, \\
& \quad \text{op} \leftarrow \text{opBar},
\end{aligned}$$

$arg \leftarrow argBar,$
 $rval \leftarrow rvalBar,$
 $tstate \leftarrow tstateBar,$
 $fate \leftarrow fateBar,$
 $to \leftarrow ord.to,$
 $tenv \leftarrow tenvBar,$
 $benv \leftarrow ord.benv,$
 $ff \leftarrow ffBar,$
 $Vinit \leftarrow V0$

$SerSpec \stackrel{\Delta}{=} Ser!SpecD$
