MODULE *MVCC*

EXTENDS *Naturals*, *TransitiveClosure*

CONSTANTS *Tr*, *Obj*, *Val*, *T*0, *V*0,
Unstarted, *Open*, *Committed*, *Aborted*,
Ok, *Err*,
Busy

ASSUME $V0 \in Val$

$None \triangleq$ CHOOSE $n : n \notin Nat$

VARIABLES

externally visible variables

op,	operation
arg,	operation argument
rval,	operation return value
tr,	transaction

internal variables

db,	database: contains object versions
vis,	set of transactions visible to each transaction
tstate,	state of each transaction
tid,	transaction id: logical timestamp of each transaction
deadlocked	transactions that have deadlocked

$SnapInit \triangleq [obj \in Obj \mapsto V0]$

$$
\begin{aligned}
TypeOk \triangleq\ & \land op \in [Tr \to \{\text{“-”}, \text{“s”}, \text{“r”}, \text{“w”}, \text{“c”}, \text{“a”}\}] \\
& \land arg \in [Tr \to \{\langle\rangle\} \cup Obj \cup Obj \times Val] \\
& \land rval \in [Tr \to Val \cup \{Ok, Busy, Err\}] \\
& \land db \in [Obj \to \text{SUBSET } [val : Val, tr : Tr]] \\
& \land vis \in [Tr \to \text{SUBSET } Tr] \\
& \land tid \in [Tr \to Nat \cup \{None\}] \\
& \land tstate \in [Tr \to \{Unstarted, Open, Committed, Aborted\}] \\
& \land deadlocked \subseteq Tr
\end{aligned}
$$

$$
\begin{aligned}
Init \triangleq\ & \land op = [t \in Tr \mapsto \text{“-”}] \\
& \land arg = [t \in Tr \mapsto \langle\rangle] \\
& \land rval = [t \in Tr \mapsto Ok] \\
& \land tr = T0 \\
& \land db = [obj \in Obj \mapsto \{[val \mapsto V0, tr \mapsto T0]\}] \\
& \land vis = [t \in Tr \mapsto \{\}] \\
& \land tid = [t \in Tr \mapsto \text{IF } t = T0 \text{ THEN } 0 \text{ ELSE } None]
\end{aligned}
$$

$\land\ tstate = [t \in Tr \mapsto \text{IF } t = T0 \text{ THEN } Committed \text{ ELSE } Unstarted]$
$\land\ deadlocked = \{\}$

Maximum value of a set

$Max(S) \triangleq \text{CHOOSE } x \in S : \forall\, y \in S \setminus \{x\} : x \geq y$

Committed transactions

$CTs \triangleq \{t \in Tr : \land\ tstate[t] = Committed\}$

Maximum transaction id

$mxid \triangleq Max(\{tid[t] : t \in Tr\} \setminus \{None\})$

$StartTransaction(t) \triangleq$
$\quad \land\ tstate[t] = Unstarted$
$\quad \land\ op' = [op \text{ EXCEPT } ![t] = \text{“s”}]$
$\quad \land\ arg' = [arg \text{ EXCEPT } ![t] = \langle\rangle]$
$\quad \land\ rval' = [rval \text{ EXCEPT } ![t] = Ok]$
$\quad \land\ tr' = t$
$\quad \land\ vis' = [vis \text{ EXCEPT } ![t] = CTs \cup \{t\}]$
$\quad \land\ tid' = [tid \text{ EXCEPT } ![t] = mxid + 1]$
$\quad \land\ tstate' = [tstate \text{ EXCEPT } ![t] = Open]$
$\quad \land \text{ UNCHANGED } \langle db,\ deadlocked\rangle$

$BeginRd(t,\ obj) \triangleq \land\ tstate[t] = Open$
$\qquad\qquad\qquad\qquad\ \land\ rval[t] \neq Busy$
$\qquad\qquad\qquad\qquad\ \land\ op' = [op \text{ EXCEPT } ![t] = \text{“r”}]$
$\qquad\qquad\qquad\qquad\ \land\ arg' = [arg \text{ EXCEPT } ![t] = obj]$
$\qquad\qquad\qquad\qquad\ \land\ rval' = [rval \text{ EXCEPT } ![t] = Busy]$
$\qquad\qquad\qquad\qquad\ \land\ tr' = t$
$\qquad\qquad\qquad\qquad\ \land \text{ UNCHANGED } \langle db,\ vis,\ tstate,\ tid,\ deadlocked\rangle$

Retrieve the version for *obj* given the set of visible transactions *vist*

$GetVer(obj,\ vist) \triangleq \text{CHOOSE } v \in db[obj] :$
$\quad \land\ v.tr \in vist$
$\quad \land \neg\exists\, w \in db[obj] : \land\ w \in db[obj]$
$\qquad\qquad\qquad\qquad\qquad\ \land\ w.tr \in vist$
$\qquad\qquad\qquad\qquad\qquad\ \land\ tid[w.tr] > tid[v.tr]$

$Get(t,\ obj) \triangleq GetVer(obj,\ vis[t]).val$

$EndRd(t,\ obj,\ val) \triangleq \land\ op[t] = \text{“r”}$
$\qquad\qquad\qquad\qquad\ \land\ rval[t] = Busy$
$\qquad\qquad\qquad\qquad\ \land\ arg[t] = obj$
$\qquad\qquad\qquad\qquad\ \land\ val = Get(t,\ obj)$
$\qquad\qquad\qquad\qquad\ \land\ tr' = t$
$\qquad\qquad\qquad\qquad\ \land\ rval' = [rval \text{ EXCEPT } ![t] = val]$
$\qquad\qquad\qquad\qquad\ \land \text{ UNCHANGED } \langle op,\ arg,\ db,\ db,\ vis,\ tstate,\ tid,\ deadlocked\rangle$

$$BeginWr(t,\ obj,\ val) \triangleq \begin{array}{l} \land\ tstate[t] = Open \\ \land\ rval[t] \neq Busy \\ \land\ op' = [op \text{ EXCEPT } ![t] = \text{“w”}] \\ \land\ arg' = [arg \text{ EXCEPT } ![t] = \langle obj,\ val\rangle] \\ \land\ rval' = [rval \text{ EXCEPT } ![t] = Busy] \\ \land\ tr' = t \\ \land \text{UNCHANGED } \langle db,\ vis,\ tid,\ tstate,\ deadlocked\rangle \end{array}$$

True if transaction t is active and has modified object obj

$$ActiveWrite(t,\ obj) \triangleq \begin{array}{l} \land\ tstate[t] = Open \\ \land \exists\, ver \in db[obj] : ver.tr = t \end{array}$$

Dependencies due to active writes

$$\begin{array}{l} Deps \triangleq \{\langle Ti,\ Tj\rangle \in Tr \times Tr : \\ \quad \land\ Ti \neq Tj \\ \quad \land\ tstate[Ti] = Open \\ \quad \land\ op[Ti] = \text{“w”} \\ \quad \land\ rval[Ti] = Busy \\ \quad \land \exists\, obj \in Obj : arg[Ti][1] = obj \land ActiveWrite(Tj,\ obj)\} \end{array}$$

Detect if deadlock is currently occurring. This only fires if there are as-yet-undetected deadlocks

$$\begin{array}{l} DetectDeadlock \triangleq \\ \quad \text{LET } TCD \triangleq TC(Deps) \\ \qquad stuck \triangleq \{t \in Tr : \langle t,\ t\rangle \in TCD\} \\ \quad \text{IN} \quad \land\ stuck \setminus deadlocked \neq \{\} \text{ something is stuck that isn't in the deadlocked set yet} \\ \qquad \land\ deadlocked' = deadlocked \cup stuck \\ \qquad \land \text{UNCHANGED } \langle op,\ arg,\ rval,\ tr,\ db,\ vis,\ tstate,\ tid\rangle \end{array}$$

True if transaction t is committed and has modified object obj

$$CommittedWrite(t,\ obj) \triangleq \begin{array}{l} \land\ tstate[t] = Committed \\ \land \exists\, ver \in db[obj] : ver.tr = t \end{array}$$

Two transactions are concurrent if neither is visible to the other

$$Concurrent(t1,\ t2) \triangleq \begin{array}{l} \land\ t1 \notin vis[t2] \\ \land\ t2 \notin vis[t1] \end{array}$$

True if there is another transaction that has a write conflict with transaction t with object obj

$$WriteConflict(t,\ obj) \triangleq \exists\, tt \in Tr \setminus \{t\} : CommittedWrite(tt,\ obj) \land Concurrent(t,\ tt)$$

$$\begin{array}{l} EndWr(t,\ obj,\ val) \triangleq \text{LET } oldwrites \triangleq \{v \in db[obj] : v.tr = t\} \\ \qquad\qquad\qquad\qquad\qquad\quad ver \triangleq [val \mapsto val,\ tr \mapsto t] \\ \qquad\qquad\qquad\qquad\quad \text{IN} \\ \qquad\qquad\qquad\qquad\quad \land\ op[t] = \text{“w”} \\ \qquad\qquad\qquad\qquad\quad \land\ arg[t] = \langle obj,\ val\rangle \end{array}$$

$$
\begin{array}{l}
\land\ rval[t] = Busy \\
\land\ \neg\exists\ tt \quad \in Tr \setminus \{t\} : \lor\ ActiveWrite(tt,\ obj) \\
\land\ \neg\ WriteConflict(t,\ obj) \\
\land\ db' = [db\ \text{EXCEPT}\ ![obj] = (@ \setminus oldwrites) \cup \{ver\}] \\
\land\ rval' = [rval\ \text{EXCEPT}\ ![t] = Ok] \\
\land\ tr' = t \\
\land\ \text{UNCHANGED}\ \ \langle op,\ arg,\ tstate,\ tid,\ vis,\ deadlocked\rangle
\end{array}
$$

$$
\begin{array}{lll}
AbortWr(t,\ obj,\ val) & \triangleq & \land\ op[t] = \text{“w”} \\
& & \land\ rval[t] = Busy \\
& & \land\ \lor\ WriteConflict(t,\ obj) \\
& & \quad \lor\ t \in deadlocked \\
& & \land\ op' = [op\ \text{EXCEPT}\ ![t] = \text{“a”}] \\
& & \land\ arg' = [arg\ \text{EXCEPT}\ ![t] = \langle\rangle] \\
& & \land\ rval' = [rval\ \text{EXCEPT}\ ![t] = Err] \\
& & \land\ tr' = t \\
& & \land\ tstate' = [tstate\ \text{EXCEPT}\ ![t] = Aborted] \\
& & \land\ \text{UNCHANGED}\ \ \langle db,\ vis,\ tid,\ deadlocked\rangle
\end{array}
$$

$$
\begin{array}{lll}
Abort(t) & \triangleq & \land\ tstate[t] = Open \\
& & \land\ rval[t] \neq Busy \\
& & \land\ op' = [op\ \text{EXCEPT}\ ![t] = \text{“a”}] \\
& & \land\ arg' = [arg\ \text{EXCEPT}\ ![t] = \langle\rangle] \\
& & \land\ rval' = [rval\ \text{EXCEPT}\ ![t] = Ok] \\
& & \land\ tr' = t \\
& & \land\ tstate' = [tstate\ \text{EXCEPT}\ ![t] = Aborted] \\
& & \land\ \text{UNCHANGED}\ \langle db,\ vis,\ tid,\ deadlocked\rangle
\end{array}
$$

$$
\begin{array}{lll}
Commit(t) & \triangleq & \land\ tstate[t] = Open \\
& & \land\ rval[t] \neq Busy \\
& & \land\ tstate' = [tstate\ \text{EXCEPT}\ ![t] = Committed] \\
& & \land\ op' = [op\ \text{EXCEPT}\ ![t] = \text{“c”}] \\
& & \land\ arg' = [arg\ \text{EXCEPT}\ ![t] = \langle\rangle] \\
& & \land\ rval' = [rval\ \text{EXCEPT}\ ![t] = Ok] \\
& & \land\ tr' = t \\
& & \land\ tstate' = [tstate\ \text{EXCEPT}\ ![t] = Committed] \\
& & \land\ \text{UNCHANGED}\ \langle db,\ vis,\ tid,\ deadlocked\rangle
\end{array}
$$

$$Done \triangleq \forall\ t \in Tr : tstate[t] \in \{Committed,\ Aborted\}$$

$$v \triangleq \langle op,\ arg,\ rval,\ tr,\ db,\ tstate,\ tid,\ vis,\ deadlocked\rangle$$

$$Termination \triangleq Done \land \text{UNCHANGED}\ v$$

$$
\begin{array}{lll}
\mathit{Next} & \triangleq & \lor \exists\, t \in \mathit{Tr},\ \mathit{obj} \in \mathit{Obj},\ \mathit{val} \in \mathit{Val} : \\
& & \quad \lor \mathit{StartTransaction}(t) \\
& & \quad \lor \mathit{BeginRd}(t,\ \mathit{obj}) \\
& & \quad \lor \mathit{EndRd}(t,\ \mathit{obj},\ \mathit{val}) \\
& & \quad \lor \mathit{BeginWr}(t,\ \mathit{obj},\ \mathit{val}) \\
& & \quad \lor \mathit{EndWr}(t,\ \mathit{obj},\ \mathit{val}) \\
& & \quad \lor \mathit{AbortWr}(t,\ \mathit{obj},\ \mathit{val}) \\
& & \quad \lor \mathit{Commit}(t) \\
& & \quad \lor \mathit{Abort}(t) \\
& & \lor \mathit{DetectDeadlock} \\
& & \lor \mathit{Termination}
\end{array}
$$

$$
\begin{array}{lll}
L & \triangleq & \land \mathrm{WF}_v(\exists\, t \in \mathit{Tr},\ \mathit{obj} \in \mathit{Obj},\ \mathit{val} \in \mathit{Val} : \\
& & \qquad \lor \mathit{EndRd}(t,\ \mathit{obj},\ \mathit{val}) \\
& & \qquad \lor \mathit{EndWr}(t,\ \mathit{obj},\ \mathit{val}) \\
& & \qquad \lor \mathit{AbortWr}(t,\ \mathit{obj},\ \mathit{val})) \\
& & \land \mathrm{WF}_v(\exists\, t \in \mathit{Tr} : \mathit{StartTransaction}(t)) \\
& & \land \mathrm{SF}_v(\exists\, t \in \mathit{Tr} : \mathit{Commit}(t) \lor \mathit{Abort}(t)) \\
& & \land \mathrm{WF}_v(\mathit{DetectDeadlock})
\end{array}
$$

$$
\mathit{Spec} \triangleq \mathit{Init} \land \Box[\mathit{Next}]_v \land L
$$