```
┌──────────────────── MODULE MVCCRefinement ────────────────────┐
EXTENDS MVCC, Naturals, Sequences, FiniteSets, TLC

CONSTANTS NULL, Flip, Flop

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

 Refinement transactions
TrR ≜ Tr \ {T0}

 Committed transactions
CT ≜ {t ∈ TrR : tstate[t] = Committed}

N ≜ Cardinality(CT)

TypeOkR ≜ ∧ TypeOk
          ∧ ∀ i ∈ DOMAIN h : LET e ≜ h[i]IN
               ∧ e.tr ∈ TrR
               ∧ e.op ∈ {"r", "w", "c", "a"}
               ∧ e.arg ∈ CASE e.op = "r" → Obj
                           □  e.op = "w" → Obj × Val
                           □  OTHER      → {⟨⟩}

               ∧ e.rval ∈ Val ∪ {Ok, Err}
               ∧ e.tstate ∈ [Tr → {Unstarted, Open, Committed, Aborted}]
               ∧ e.op ∈ {"r", "w"} ⇒ ∧ DOMAIN e.wr ⊆ Obj
                                     ∧ ∀ obj ∈ DOMAIN e.wr : e.wr[obj] ∈ Val

          ∧ fateIsSet ∈ BOOLEAN
          ∧ canIssue ∈ BOOLEAN
          ∧ parity ∈ {0, 1}
          ∧ reads ∈ [Tr → SUBSET Obj]
          ∧ writes ∈ [Tr → SUBSET Obj]
          ∧ tenvBar ∈ [CT → [Obj → Val]] ∪ {NULL}
          ∧ ord ∈ [to : [1 . . N → CT] ∪ {NULL}, benv : [1 . . N + 1 → [Obj → Val]] ∪ {NULL}]

InitR ≜ ∧ Init
        ∧ fateIsSet = FALSE
        ∧ parity = 0
        ∧ h = ⟨⟩
        ∧ canIssue = FALSE
        ∧ reads = [t ∈ Tr ↦ {}]
        ∧ writes = [t ∈ Tr ↦ IF t = T0 THEN Obj ELSE {}]
        ∧ ord = [to ↦ NULL, benv ↦ NULL]
        ∧ tenvBar = NULL

StartTransactionR(t) ≜ ∧ StartTransaction(t)
                       ∧ UNCHANGED ⟨h, fateIsSet, canIssue, parity, reads, writes, ord, tenvBar⟩
```

$$\begin{aligned}\mathit{BeginRdR}(t,\ \mathit{obj}) \triangleq\ & \land\ \mathit{BeginRd}(t,\ \mathit{obj})\\ & \land\ \textsc{unchanged}\ \langle h,\ \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{parity},\ \mathit{reads},\ \mathit{writes},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

$$\begin{aligned}&\mathit{EndRdR}(t,\ \mathit{obj},\ \mathit{val}) \triangleq\\ &\quad \land\ \mathit{EndRd}(t,\ \mathit{obj},\ \mathit{val})\\ &\quad \land\ h' = \mathit{Append}(h,\ [\mathit{tr} \mapsto t,\ \mathit{op} \mapsto \text{“r”},\ \mathit{arg} \mapsto \mathit{obj},\ \mathit{rval} \mapsto \mathit{val},\\ &\qquad\qquad\qquad\qquad \mathit{tstate} \mapsto \mathit{tstate},\ \mathit{wr} \mapsto [o \in \mathit{writes}[t] \mapsto \mathit{Get}(t,\ o)]])\\ &\quad \land\ \mathit{reads}' = \textsc{if}\ \mathit{obj} \in \mathit{writes}[t]\ \textsc{then}\ \mathit{reads}\ \textsc{else}\ [\mathit{reads}\ \textsc{except}\ ![t] = @ \cup \{\mathit{obj}\}] \quad \text{unwritten reads}\\ &\quad \land\ \mathit{parity}' = 1 - \mathit{parity}\\ &\quad \land\ \textsc{unchanged}\ \langle \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{writes},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

$$\begin{aligned}\mathit{BeginWrR}(t,\ \mathit{obj},\ \mathit{val}) \triangleq\ & \land\ \mathit{BeginWr}(t,\ \mathit{obj},\ \mathit{val})\\ & \land\ \textsc{unchanged}\ \langle h,\ \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{parity},\ \mathit{reads},\ \mathit{writes},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

$$\begin{aligned}&\mathit{EndWrR}(t,\ \mathit{obj},\ \mathit{val}) \triangleq\\ &\quad \land\ \mathit{EndWr}(t,\ \mathit{obj},\ \mathit{val})\\ &\quad \land\ h' = \mathit{Append}(h,\ [\mathit{tr} \mapsto t,\ \mathit{op} \mapsto \text{“w”},\ \mathit{arg} \mapsto \langle \mathit{obj},\ \mathit{val}\rangle,\ \mathit{rval} \mapsto \mathit{Ok},\\ &\qquad\qquad\qquad\qquad \mathit{tstate} \mapsto \mathit{tstate},\ \mathit{wr} \mapsto [o \in \mathit{writes}[t] \mapsto \mathit{Get}(t,\ o)]])\\ &\quad \land\ \mathit{writes}' = [\mathit{writes}\ \textsc{except}\ ![t] = @ \cup \{\mathit{obj}\}]\\ &\quad \land\ \mathit{parity}' = 1 - \mathit{parity}\\ &\quad \land\ \textsc{unchanged}\ \langle \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{reads},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

$$\begin{aligned}&\mathit{AbortWrR}(t,\ \mathit{obj}) \triangleq\\ &\quad \land\ \mathit{AbortWr}(t,\ \mathit{obj})\\ &\quad \land\ h' = \mathit{Append}(h,\ [\mathit{tr} \mapsto t,\ \mathit{op} \mapsto \text{“a”},\ \mathit{arg} \mapsto \langle\rangle,\ \mathit{rval} \mapsto \mathit{Err},\\ &\qquad\qquad\qquad\qquad \mathit{tstate} \mapsto [\mathit{tstate}\ \textsc{except}\ ![t] = \mathit{Aborted}]])\\ &\quad \land\ \textsc{unchanged}\ \langle \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{parity},\ \mathit{reads},\ \mathit{writes},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

$$\begin{aligned}&\mathit{CommitR}(t) \triangleq\\ &\quad \land \qquad \mathit{Commit}(t)\\ &\quad \land \qquad h' = \mathit{Append}(h,\ [\mathit{tr} \mapsto t,\ \mathit{op} \mapsto \text{“c”},\ \mathit{arg} \mapsto \langle\rangle,\ \mathit{rval} \mapsto \mathit{Ok},\\ &\qquad\qquad\qquad\qquad\qquad \mathit{tstate} \mapsto [\mathit{tstate}\ \textsc{except}\ ![t] = \mathit{Committed}]])\\ &\quad \land \qquad \textsc{unchanged}\ \langle \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{parity},\ \mathit{reads},\ \mathit{writes},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

$$\begin{aligned}&\mathit{AbortR}(t) \triangleq\\ &\quad \land\ \mathit{Abort}(t)\\ &\quad \land\ h' = \mathit{Append}(h,\ [\mathit{tr} \mapsto t,\ \mathit{op} \mapsto \text{“a”},\ \mathit{arg} \mapsto \langle\rangle,\ \mathit{rval} \mapsto \mathit{Ok},\\ &\qquad\qquad\qquad\qquad \mathit{tstate} \mapsto [\mathit{tstate}\ \textsc{except}\ ![t] = \mathit{Aborted}]])\\ &\quad \land\ \textsc{unchanged}\ \langle \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{parity},\ \mathit{reads},\ \mathit{writes},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

$$\begin{aligned}\mathit{DetectDeadlockR} \triangleq\ & \land\ \mathit{DetectDeadlock}\\ & \land\ \textsc{unchanged}\ \langle h,\ \mathit{fateIsSet},\ \mathit{canIssue},\ \mathit{parity},\ \mathit{reads},\ \mathit{writes},\ \mathit{ord},\ \mathit{tenvBar}\rangle\end{aligned}$$

Get the order in which this transactionruns

$$\mathit{Ord}(t) \triangleq \textsc{choose}\ i \in \textsc{domain}\ \mathit{ord.to} : \mathit{ord.to}[i] = t$$

$$\begin{aligned}\mathit{SetFate} \triangleq\ & \land\ \mathit{Done}\\ & \land\ \mathit{fateIsSet} = \textsc{false}\end{aligned}$$

```
        $\land\ fateIsSet' = \text{TRUE}$
        $\land\ ord' = \text{CHOOSE}\ r \in [to : [1\,..\,N \to CT],\ benv : [1\,..\,N+1 \to [Obj \to Val]]] :$
            first environment must be the initialization
            $\land\ r.benv[1] = SnapInit$
            to must be a total ordering
            $\land\ \forall\, i, j \in 1\,..\,N : r.to[i] = r.to[j] \Rightarrow i = j$
            $\land\ \forall\, i \in 1\,..\,N : \text{LET}\ t \triangleq r.to[i]\text{IN}$
                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
                $\land\ \forall\, obj \in Obj : (r.benv[i+1][obj] \neq r.benv[i][obj]) \Rightarrow obj \in writes[t]$
        $\land\ tenvBar' = \text{LET}\ ordp \triangleq ord'$
                           $benv \triangleq ordp.benv$
                           $to \triangleq ordp.to\text{IN}$
            $[t \in CT \mapsto \text{LET}\ i \triangleq \text{CHOOSE}\ i \in \text{DOMAIN}\ to : to[i] = t\text{IN}\quad benv[i]]$
        $\land\ \text{UNCHANGED}\ \langle op,\ arg,\ rval,\ tr,\ db,\ vis,\ tstate,\ tid,\ deadlocked,$
                       $h,\ canIssue,\ parity,\ reads,\ writes\rangle$

$Issue \triangleq$ $\land\ h \neq \langle\rangle$
       $\land\ fateIsSet$
       $\land\ canIssue' = \text{TRUE}$
       $\land\ h' = \text{IF}\ canIssue\ \text{THEN}\ Tail(h)\ \text{ELSE}\ h$
       $\land\ h' \neq \langle\rangle$
       $tenvBar'$ needs to reflect the state of the *next* head in the history, not the current head
       $\land\ tenvBar' = \text{LET}\ e \triangleq Head(h')$
                         $obj \triangleq e.arg[1]$
                         $val \triangleq e.arg[2]$
                         $t \triangleq e.tr$
                     $\text{IN}$   $\text{IF}\ tstate[e.tr] = Committed \land e.op = \text{“w”}$
                          $\text{THEN}\ [tenvBar\ \text{EXCEPT}\ ![t][obj] = val]$
                          $\text{ELSE}\ tenvBar$
       $\land\ \text{UNCHANGED}\ \langle op,\ arg,\ rval,\ tr,\ db,\ vis,\ tstate,\ tid,\ deadlocked,$
                      $fateIsSet,\ parity,\ reads,\ writes,\ ord\rangle$

$vv \triangleq \langle op,\ arg,\ rval,\ tr,\ db,\ vis,\ tstate,\ tid,\ deadlocked,\ h,\ fateIsSet,\ canIssue,$
       $parity,\ reads,\ writes,\ ord,\ tenvBar\rangle$

$TerminationR \triangleq$ $\land\ Done$
              $\land\ Tail(h) = \langle\rangle$
              $\land\ \text{UNCHANGED}\ vv$

$NextR \triangleq$ $\lor\ \exists\, t \in Tr,\ obj \in Obj,\ val \in Val :$
          $\lor\ StartTransactionR(t)$
```

```
            ∨ BeginRdR(t, obj)
            ∨ EndRdR(t, obj, val)
            ∨ BeginWrR(t, obj, val)
            ∨ EndWrR(t, obj, val)
            ∨ AbortWrR(t, obj)
            ∨ CommitR(t)
            ∨ AbortR(t)
          ∨ DetectDeadlockR
          ∨ Issue
          ∨ SetFate
          ∨ TerminationR

SpecR ≜ InitR ∧ □[NextR]_vv

trBar  ≜ IF canIssue THEN Head(h).tr ELSE T0
opBar ≜ IF canIssue THEN Head(h).op ELSE "r"
argBar ≜ CASE canIssue ∧ Head(h).arg = ⟨⟩ → None
            □  canIssue ∧ Head(h).arg ≠ ⟨⟩ → Head(h).arg
            □  OTHER  → CHOOSE obj ∈ Obj : TRUE
rvalBar ≜ CASE canIssue ∧ Head(h).rval ≠ Err → Head(h).rval
             □  canIssue ∧ Head(h).rval = Err → Ok
             □  OTHER                          → V0
tstateBar ≜ [t ∈ TrR ↦
               LET s ≜ Head(h).tstate[t]IN
               CASE ¬canIssue                → Open
                 □  canIssue ∧ s = Unstarted  → Open
                 □  canIssue ∧ s = Open       → Open
                 □  canIssue ∧ s = Committed → Committed
                 □  canIssue ∧ s = Aborted    → Aborted]

ffBar ≜ LET Parity(hh) ≜ Len(SelectSeq(hh, LAMBDA e : e.op ∈ {"r", "w"}))%2
            p ≜ Parity(h)
            opp ≜ Head(h).opIN
   CASE ¬canIssue                          → Flip
     □  canIssue ∧ opp   ∈ {"r", "w"} ∧ parity = p → Flop
     □  canIssue ∧ opp ∉ {"r", "w"}   ∧ parity = p → Flip
     □  canIssue ∧ opp   ∈ {"r", "w"} ∧ parity ≠ p → Flip
     □  canIssue ∧ opp ∉ {"r", "w"}   ∧ parity ≠ p → Flop

fateBar ≜ IF ¬fateIsSet THEN NULL
            ELSE [t ∈ TrR ↦ tstate[t]]

Ser ≜ INSTANCE SerializabilityD WITH
   Tr ← TrR,
   tr  ← trBar,
   op ← opBar,
```

$$
\begin{array}{l}
\quad \mathit{arg} \leftarrow \mathit{argBar}, \\
\quad \mathit{rval} \leftarrow \mathit{rvalBar}, \\
\quad \mathit{tstate} \leftarrow \mathit{tstateBar}, \\
\quad \mathit{fate} \leftarrow \mathit{fateBar}, \\
\quad \mathit{to} \leftarrow \mathit{ord.to}, \\
\quad \mathit{tenv} \leftarrow \mathit{tenvBar}, \\
\quad \mathit{benv} \leftarrow \mathit{ord.benv}, \\
\quad \mathit{ff} \leftarrow \mathit{ffBar}, \\
\quad \mathit{Vinit} \leftarrow V0
\end{array}
$$

$\mathit{SerSpec} \;\triangleq\; \mathit{Ser}\,!\,\mathit{SpecD}$

THEOREM $\mathit{SpecR} \Rightarrow \mathit{SerSpec}$