MODULE *Serializability*

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

CONSTANTS Tr, Obj, Val,
$Open$, $Committed$, $Aborted$,
Ok,
$Flip$, $Flop$,
$Vinit$, $T0$, $None$

To check refinement in TLC, we need to specify these as constants
$T0 \triangleq$ CHOOSE $t : t \notin Tr$
$None \triangleq$ CHOOSE $v : v \notin Obj \cup (Obj \times Val) \cup Pred$

VARIABLES
externally visible variables
op, operation
arg, operation argument
$rval$, operation return value
tr, transaction

internal variables
$tstate$, state of transaction (open, committed, aborted)
$fate$, the ultimate fate of each transaction:
committed or aborted
to, transaction order: a sequence that indicates
the commit order of committed transactions
$tenv$, value of variables for each transaction
$benv$, sequence: beginning state of the i'th transaction
ff flip/flop

$v \triangleq \langle tr, op, arg, rval, tstate, fate, to, tenv, benv, ff \rangle$

committed transactions
$CT \triangleq \{t \in Tr : fate[t] = Committed\}$

$N \triangleq Cardinality(CT)$

Generate all permuted sequences of the set S
$Orderings(S) \triangleq \{seq \in [1 \,..\, Cardinality(S) \rightarrow S] : \forall\, i, j \in \text{DOMAIN } seq : seq[i] = seq[j] \Rightarrow i = j\}$

the ordinal value ($e.g.$, 1,2,3) of a committed transaction
$Ord(t) \triangleq$ CHOOSE $i \in$ DOMAIN $to : to[i] = t$

$Toggle(f) \triangleq$ CASE $f = Flip \rightarrow Flop$
$\Box\ f = Flop \rightarrow Flip$

$$
\begin{aligned}
Init \triangleq\; & \land\, tr = T0 \\
& \land\, op = \text{“r”} \\
& \land\, arg \in Obj \\
& \land\, rval = Vinit \\
& \land\, tstate = [t \in Tr \mapsto Open] \\
& \land\, fate \in [Tr \rightarrow \{Committed, Aborted\}] \\
& \land\, to \in Orderings(CT) \\
& \land\, benv \in [1 \,..\, N+1 \rightarrow [Obj \rightarrow Val]] \\
& \land\, tenv \in \{f \in [CT \rightarrow [Obj \rightarrow Val]] : \forall\, t \in CT : f[t] = benv[Ord(t)]\} \\
& \land\, ff \in \{Flip, Flop\}
\end{aligned}
$$

$$
\begin{aligned}
Read(t, obj, val) \triangleq\; & \land\, tstate[t] = Open \\
& \land\, \lor\, fate[t] = Aborted \quad \text{for aborted commits, we don't care what the read value is} \\
& \quad\;\, \lor\, fate[t] = Committed \land val = tenv[t][obj] \\
& \land\, tr' = t \\
& \land\, op' = \text{“r”} \\
& \land\, arg' = obj \\
& \land\, rval' = val \\
& \land\, ff' = Toggle(ff) \\
& \land\, \text{UNCHANGED } \langle tstate, fate, to, tenv, benv \rangle
\end{aligned}
$$

$$
\begin{aligned}
Write(t, obj, val) \triangleq\; & \land\, tstate[t] = Open \\
& \land\, tr' = t \\
& \land\, op' = \text{“w”} \\
& \land\, arg' = \langle obj, val \rangle \\
& \land\, rval' = Ok \\
& \land\, tenv' = \text{IF } fate[t] = Committed \text{ THEN } [tenv \text{ EXCEPT } ![t][obj] = val] \text{ ELSE } tenv \\
& \land\, ff' = Toggle(ff) \\
& \land\, \text{UNCHANGED } \langle tstate, fate, to, benv \rangle
\end{aligned}
$$

$$
\begin{aligned}
Commit(t) \triangleq\; & \land\, tstate[t] = Open \\
& \land\, fate[t] = Committed \\
& \land\, tenv[t] = benv[Ord(t) + 1] \\
& \land\, tr' = t \\
& \land\, op' = \text{“c”} \\
& \land\, arg' = None \\
& \land\, rval' = Ok \\
& \land\, tstate' = [tstate \text{ EXCEPT } ![t] = Committed] \\
& \land\, \text{UNCHANGED } \langle fate, to, tenv, benv, ff \rangle
\end{aligned}
$$

$$
\begin{aligned}
Abort(t) \triangleq\; & \land\, tstate[t] = Open \\
& \land\, fate[t] = Aborted \\
& \land\, tr' = t \\
& \land\, op' = \text{“a”} \\
& \land\, arg' = None
\end{aligned}
$$

$$\begin{aligned}
&\land\ rval' = Ok\\
&\land\ tstate' = [tstate \text{ EXCEPT } ![t] = Aborted]\\
&\land\ \text{UNCHANGED } \langle fate,\ to,\ tenv,\ benv,\ ff \rangle
\end{aligned}$$

$$\begin{aligned}
Termination \triangleq\ &\land \forall\, t \in Tr : tstate[t] \in \{Committed,\ Aborted\}\\
&\land\ \text{UNCHANGED } v
\end{aligned}$$

$$\begin{aligned}
Next \triangleq\ &\lor \exists\, t \in Tr :\\
&\qquad \lor\ Commit(t)\\
&\qquad \lor\ Abort(t)\\
&\qquad \lor \exists\, obj \in Obj,\ val \in Val :\\
&\qquad\qquad \lor\ Read(t,\ obj,\ val)\\
&\qquad\qquad \lor\ Write(t,\ obj,\ val)\\
&\lor\ Termination
\end{aligned}$$

Number of variables with the same values in environments $e1$ and $e2$

$$M(e1,\ e2) \triangleq Cardinality(\{obj \in Obj : e1[obj] = e2[obj]\})$$

$W(j,\ k)$ is true if there's a transaction t doing a write where:

1. the number of variables in the $1st$ state that are equal to the expected values is j
2. the number of variables in the $2nd$ state that are equal to the expected values is k

$$\begin{aligned}
W(j,\ k) \triangleq\ &\exists\, t \in CT,\ obj \in Obj,\ val \in Val :\\
&\quad \land\ Write(t,\ obj,\ val)\\
&\quad \land\ M(tenv[t],\ \ benv[Ord(t) + 1]) = j\\
&\quad \land\ M(tenv'[t],\ benv[Ord(t) + 1]) = k
\end{aligned}$$

$$\begin{aligned}
L \triangleq\ &\land\ \mathrm{WF}_v(\exists\, t \in Tr : Abort(t))\\
&\land\ \mathrm{SF}_v(\exists\, t\ \in Tr : Commit(t))\\
&\land\ \mathrm{WF}_v(W(0,\ 1))\\
&\land \forall\, i \in 1 \,..\, Cardinality(Obj) - 1 : \mathrm{SF}_v(W(i,\ i + 1))
\end{aligned}$$

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