MODULE $t2pc$

EXTENDS $Integers$, $Sequences$, $FiniteSets$, TLC

CONSTANT RM,
$RMMAYFAIL$,
$TMMAYFAIL$

```
--algorithm TransactionCommit{
  variable rmState = [rm ∈ RM ↦ "working"],
           tmState = "init" ;
  define {
    canCommit ≜ ∀ rm ∈ RM : rmState[rm] ∈ {"prepared", "committed", "hidden"} ∧ tmState ≠ "abort"
    canAbort ≜ ∀ rm ∈ RM : rmState[rm] ≠ "committed" ∧ tmState ≠ "commit"
    }
  macro Prepare( p ) {
    when rmState[p] = "working" ;
    rmState[p] := "prepared" ;
    }
  macro Decide( p ) {
    either { when ∧ rmState[p] = "prepared"
                  ∧ tmState = "commit" ;
             rmState[p] := "committed" ;
           }
    or     { when ∧ rmState[p] ∈ {"working", "prepared"}
                  ∧ tmState = "abort" ;
             rmState[p] := "aborted" ;
           }
    }
   macro Fail( p ) {
   if ( RMMAYFAIL )
         {
     when rmState[p] ∈ {"working", "prepared"};
    rmState[p] := "hidden" ;
        }
    }

  fair  process ( RManager ∈ RM ) {
   RS: while ( rmState[self] ∈ {"working", "prepared"} ) {
      either Prepare(self)or Decide(self)or Fail(self) }
    }

   fair  process ( TManager = 0 ) {
    TS: either { await canCommit ;
                 TC: tmState := "commit" ;

                 F1: if ( TMMAYFAIL ) tmState := "hidden" ;
                 }
```

```
        or { await canAbort ;
             TA: tmState := "abort" ;

             F2: if ( TMMAYFAIL ) tmState := "hidden" ;
               }
    }

   fair  process ( BTManager = 1 ) {
    BS: either { await (canCommit ∧ tmState = "hidden") ∨ (∀ rm ∈ RM : rmState[rm] ∈ {"hidden",
                                                                                      "committed"}) ;
           BC: tmState := "commit" ;
            }
         or { await (canAbort ∧ tmState = "hidden") ∨ (∀ rm ∈ RM : rmState[rm] ∈ {"hidden", "aborted"})
           BA: tmState := "abort" ;
            }
    }
}
```

BEGIN TRANSLATION

VARIABLES $rmState$, $tmState$, pc

define statement

$canCommit \triangleq \forall\, rm \in RM : rmState[rm] \in \{\text{"prepared"}, \text{"committed"}, \text{"hidden"}\} \land tmState \neq \text{"abort"}$

$canAbort \triangleq \forall\, rm \in RM : rmState[rm] \neq \text{"committed"} \land tmState \neq \text{"commit"}$

$vars \triangleq \langle rmState, tmState, pc\rangle$

$ProcSet \triangleq (RM) \cup \{0\} \cup \{1\}$

$Init \triangleq$ Global variables

$\land\, rmState = [rm \in RM \mapsto \text{"working"}]$

$\land\, tmState = \text{"init"}$

$\land\, pc = [self \in ProcSet \mapsto \text{CASE } self \in RM \to \text{"RS"}$

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \square\ \ self = 0 \to \text{"TS"}$

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \square\ \ self = 1 \to \text{"BS"}]$

$RS(self) \triangleq \land\, pc[self] = \text{"RS"}$

$\qquad \land\, \text{IF } rmState[self] \in \{\text{"working"}, \text{"prepared"}\}$

$\qquad\qquad \text{THEN } \land \lor \land\, rmState[self] = \text{"working"}$

$\qquad\qquad\qquad\qquad\qquad\quad \land\, rmState' = [rmState \text{ EXCEPT } ![self] = \text{"prepared"}]$

$\qquad\qquad\qquad\qquad\qquad \lor \land \lor \land \land\, rmState[self] = \text{"prepared"}$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad \land\, tmState = \text{"commit"}$

$\qquad\qquad\qquad\qquad\qquad\qquad \land\, rmState' = [rmState \text{ EXCEPT } ![self] = \text{"committed"}]$

$\qquad\qquad\qquad\qquad\qquad\quad \lor \land \land\, rmState[self] \in \{\text{"working"}, \text{"prepared"}\}$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad \land\, tmState = \text{"abort"}$

$\qquad\qquad\qquad\qquad\qquad\qquad \land\, rmState' = [rmState \text{ EXCEPT } ![self] = \text{"aborted"}]$

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\lor\ \land\ \text{IF}\ \mathit{RMMAYFAIL} \\
\qquad\qquad\qquad\qquad\qquad\quad \text{THEN}\ \land\ \mathit{rmState}' = [\mathit{rmState}\ \text{EXCEPT}\ ![\mathit{self}] = \text{"hidden"}] \\
\qquad\qquad\qquad\qquad\qquad\quad \text{ELSE}\ \ \land\ \text{TRUE} \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \ \land\ \text{UNCHANGED}\ \mathit{rmState} \\
\qquad\qquad\qquad\quad\ \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \text{"RS"}] \\
\qquad\qquad\quad \text{ELSE}\ \ \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \text{"Done"}] \\
\qquad\qquad\qquad\quad\ \land\ \text{UNCHANGED}\ \mathit{rmState} \\
\qquad\qquad \land\ \text{UNCHANGED}\ \mathit{tmState}
\end{array}
$$

$$
\mathit{RManager}(\mathit{self}) \triangleq \mathit{RS}(\mathit{self})
$$

$$
\begin{array}{lll}
\mathit{TS} & \triangleq & \land\ \mathit{pc}[0] = \text{"TS"} \\
& & \land\ \lor\ \land\ \mathit{canCommit} \\
& & \qquad\ \ \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![0] = \text{"TC"}] \\
& & \quad\ \ \lor\ \land\ \mathit{canAbort} \\
& & \qquad\ \ \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![0] = \text{"TA"}] \\
& & \land\ \text{UNCHANGED}\ \langle \mathit{rmState}, \mathit{tmState} \rangle
\end{array}
$$

$$
\begin{array}{lll}
\mathit{TC} & \triangleq & \land\ \mathit{pc}[0] = \text{"TC"} \\
& & \land\ \mathit{tmState}' = \text{"commit"} \\
& & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![0] = \text{"F1"}] \\
& & \land\ \text{UNCHANGED}\ \mathit{rmState}
\end{array}
$$

$$
\begin{array}{lll}
F1 & \triangleq & \land\ \text{IF}\ \mathit{TMMAYFAIL} \\
& & \qquad \text{THEN}\ \land\ \mathit{tmState}' = \text{"hidden"} \\
& & \qquad \text{ELSE}\ \ \land\ \text{TRUE} \\
& & \qquad\qquad\quad\ \ \land\ \text{UNCHANGED}\ \mathit{tmState} \\
& & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![0] = \text{"Done"}] \\
& & \land\ \text{UNCHANGED}\ \mathit{rmState}
\end{array}
$$

$$
\begin{array}{lll}
\mathit{TA} & \triangleq & \land\ \mathit{pc}[0] = \text{"TA"} \\
& & \land\ \mathit{tmState}' = \text{"abort"} \\
& & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![0] = \text{"F2"}] \\
& & \land\ \text{UNCHANGED}\ \mathit{rmState}
\end{array}
$$

$$
\begin{array}{lll}
F2 & \triangleq & \land\ \mathit{pc}[0] = \text{"F2"} \\
& & \land\ \text{IF}\ \mathit{TMMAYFAIL} \\
& & \qquad \text{THEN}\ \land\ \mathit{tmState}' = \text{"hidden"} \\
& & \qquad \text{ELSE}\ \ \land\ \text{TRUE} \\
& & \qquad\qquad\quad\ \ \land\ \text{UNCHANGED}\ \mathit{tmState} \\
& & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![0] = \text{"Done"}] \\
& & \land\ \text{UNCHANGED}\ \mathit{rmState}
\end{array}
$$

$$
\mathit{TManager} \triangleq \mathit{TS} \lor \mathit{TC} \lor F1 \lor \mathit{TA} \lor F2
$$

$$
\mathit{BS} \triangleq \land\ \mathit{pc}[1] = \text{"BS"}
$$

$$
\begin{array}{l}
\quad \land \lor \land (canCommit \land tmState = \text{“hidden”}) \lor (\forall\, rm \in RM : rmState[rm] \in \{\text{“hidden”},\ \text{“committed”}\}) \\
\qquad \land pc' = [pc \text{ EXCEPT } ![1] \ = \text{“BC”}] \\
\quad\ \lor \land (canAbort \land tmState = \text{“hidden”}) \lor (\forall\, rm \in RM : rmState[rm] \in \{\text{“hidden”},\ \text{“aborted”}\}) \\
\qquad \land pc' = [pc \text{ EXCEPT } ![1] = \text{“BA”}] \\
\quad \land \text{UNCHANGED } \langle rmState,\ tmState \rangle
\end{array}
$$

$$
\begin{array}{lll}
BC & \triangleq & \land pc[1] = \text{“BC”} \\
& & \land tmState' = \text{“commit”} \\
& & \land pc' = [pc \text{ EXCEPT } ![1] = \text{“Done”}] \\
& & \land \text{UNCHANGED } rmState
\end{array}
$$

$$
\begin{array}{lll}
BA & \triangleq & \land pc[1] = \text{“BA”} \\
& & \land tmState' = \text{“abort”} \\
& & \land pc' = [pc \text{ EXCEPT } ![1] = \text{“Done”}] \\
& & \land \text{UNCHANGED } rmState
\end{array}
$$

$$BTManager \triangleq BS \lor BC \lor BA$$

$$
\begin{array}{lll}
Next & \triangleq & TManager \lor BTManager \\
& & \quad \lor (\exists\, self \in RM : RManager(self)) \\
& & \quad \lor \ \text{Disjunct to prevent deadlock on termination} \\
& & \qquad ((\forall\, self \in ProcSet : pc[self] = \text{“Done”}) \land \text{UNCHANGED } vars)
\end{array}
$$

$$
\begin{array}{lll}
Spec & \triangleq & \land Init \land \Box[Next]_{vars} \\
& & \land \forall\, self \in RM : \mathrm{WF}_{vars}(RManager(self)) \\
& & \land \mathrm{WF}_{vars}(TManager) \\
& & \land \mathrm{WF}_{vars}(BTManager)
\end{array}
$$

$$Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})$$

END TRANSLATION

$$Consistent \triangleq \forall\, rm1,\ rm2 \in RM : \neg \land rmState[rm1] = \text{“aborted”} \land rmState[rm2] = \text{“committed”}$$

* Modification History
* Last modified *Tue Dec* 05 03:38:00 *EST* 2017 by *ashutoshahmadalexandar*
* Created *Wed Nov* 29 15:16:19 *EST* 2017 by *ashutoshahmadalexandar*