MODULE *t2pc*

* *ZHENG KAI* 50247576 **
* *XINBO YU* 50102922 **

```
EXTENDS $Integers, Sequences, FiniteSets, TLC$
CONSTANT $RM, RMMAYFAIL, TMMAYFAIL$

--algorithm $t2pc${
    variable $rmState = [rm \in RM \mapsto$ "working"$]$ ;
             $tmState =$ "init" ;
    define {
        $canCommit \triangleq \forall rm \in RM : rmState[rm] \in \{$"prepared", "committed"$\}$
        $canAbort \triangleq \exists rm \in RM : rmState[rm] \in \{$"aborted", "failed"$\} \land tmState \neq$ "committed"
    }

    macro $Prepare( p )$ {
        await $rmState[p] =$ "working" ;
              $rmState[p] :=$ "prepared" ;
    }

    macro $Decide( p )$ {
        either { if ( $rmState[p] =$ "prepared" $\land tmState =$ "committed" )
        {
                $rmState[p] :=$ "committed"
                }
        }
        or { if ( $rmState[p] \in \{$"working"$\} \lor ((rmState[p] =$ "prepared"$) \land (tmState =$ "aborted"$))$ )
        {
                $rmState[p] :=$ "aborted"
                }
        }
    }

    macro $Fail( p )$ {
        if ( $RMMAYFAIL$ ) {
            either $rmState[p] :=$ "failed" ;
            or skip ;
        }
    }

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

    fair process ( $TManager = 0$ ) {
        $TS$: either { await $canCommit$ ;
```

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

        or { await canAbort ;
            TA: tmState := "aborted" ;
            F2: if ( TMMAYFAIL ) tmState := "hidden" ;  }
    }

    TM backup
   fair process ( TManagerBackup = 1 ) {
       L1: await  ∨ TMMAYFAIL ∧ tmState = "hidden"
                  ∨ ¬TMMAYFAIL ;
       if ( tmState = "hidden" ) {
           TS: either {
                 await canCommit ;
                 TC: tmState := "committed" ;
               }
               or {
                 await canAbort ;
                 TA: tmState := "aborted" ;
               }
           }
       }
}
```

BEGIN TRANSLATION

Label *TS* of process *TManager* at line 48 col 13 changed to *TS_*

Label *TC* of process *TManager* at line 49 col 17 changed to *TC_*

Label *TA* of process *TManager* at line 53 col 17 changed to *TA_*

VARIABLES $rmState$, $tmState$, pc

define statement

$canCommit \triangleq \forall rm \in RM : rmState[rm] \in \{\text{"prepared"}, \text{"committed"}\}$

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

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

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

$Init \triangleq$ Global variables

$$\begin{aligned} &\land rmState = [rm \in RM \mapsto \text{"working"}] \\ &\land tmState = \text{"init"} \\ &\land pc = [self \in ProcSet \mapsto \text{CASE } self \in RM \rightarrow \text{"RS"} \\ &\qquad\qquad\qquad\qquad \Box\ self = 0 \rightarrow \text{"TS\_"} \\ &\qquad\qquad\qquad\qquad \Box\ self = 1 \rightarrow \text{"L1"}] \end{aligned}$$

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

```
∧ IF rmState[self] ∈ {"working", "prepared"}
     THEN ∧ ∨ ∧ rmState[self] = "working"
              ∧ rmState' = [rmState EXCEPT ![self] = "prepared"]
            ∨ ∧ ∨ ∧ IF rmState[self] = "prepared" ∧ tmState = "committed"
                         THEN ∧ rmState' = [rmState EXCEPT ![self] = "committed"]
                         ELSE ∧ TRUE
                              ∧ UNCHANGED rmState
                  ∨ ∧ IF rmState[self] ∈ {"working"} ∨ ((rmState[self] = "prepared") ∧ (tmSta
                         THEN ∧ rmState' = [rmState EXCEPT ![self] = "aborted"]
                         ELSE ∧ TRUE
                              ∧ UNCHANGED rmState
            ∨ ∧ IF RMMAYFAIL
                   THEN ∧ ∨ ∧ rmState' = [rmState EXCEPT ![self] = "failed"]
                          ∨ ∧ TRUE
                            ∧ UNCHANGED rmState
                   ELSE ∧ TRUE
                        ∧ UNCHANGED rmState
          ∧ pc' = [pc EXCEPT ![self] = "RS"]
     ELSE ∧ pc' = [pc EXCEPT ![self] = "Done"]
          ∧ UNCHANGED rmState
∧ UNCHANGED tmState

RManager(self) ≜ RS(self)

TS_ ≜ ∧ pc[0] = "TS_"
      ∧ ∨ ∧ canCommit
            ∧ pc' = [pc EXCEPT ![0] = "TC_"]
        ∨ ∧ canAbort
            ∧ pc' = [pc EXCEPT ![0] = "TA_"]
      ∧ UNCHANGED ⟨rmState, tmState⟩

TC_ ≜ ∧ pc[0] = "TC_"
      ∧ tmState' = "committed"
      ∧ pc' = [pc EXCEPT ![0] = "F1"]
      ∧ UNCHANGED rmState

F1 ≜ ∧ pc[0] = "F1"
     ∧ IF TMMAYFAIL
          THEN ∧ tmState' = "hidden"
          ELSE ∧ TRUE
               ∧ UNCHANGED tmState
     ∧ pc' = [pc EXCEPT ![0] = "Done"]
     ∧ UNCHANGED rmState

TA_ ≜ ∧ pc[0] = "TA_"
      ∧ tmState' = "aborted"
```

```
          $\land pc' = [pc \text{ EXCEPT } ![0] = \text{"F2"}]$
          $\land \text{UNCHANGED } rmState$

$F2 \triangleq \land pc[0] = \text{"F2"}$
         $\land \text{IF } TMMAYFAIL$
              $\text{THEN } \land tmState' = \text{"hidden"}$
              $\text{ELSE } \land \text{TRUE}$
                     $\land \text{UNCHANGED } tmState$
         $\land pc' = [pc \text{ EXCEPT } ![0] = \text{"Done"}]$
         $\land \text{UNCHANGED } rmState$

$TManager \triangleq TS\_ \lor TC\_ \lor F1 \lor TA\_ \lor F2$

$L1 \triangleq \land pc[1] = \text{"L1"}$
         $\land \lor TMMAYFAIL \land tmState = \text{"hidden"}$
           $\lor \neg TMMAYFAIL$
         $\land \text{IF } tmState = \text{"hidden"}$
              $\text{THEN } \land pc' = [pc \text{ EXCEPT } ![1] = \text{"TS"}]$
              $\text{ELSE } \land pc' = [pc \text{ EXCEPT } ![1] = \text{"Done"}]$
         $\land \text{UNCHANGED } \langle rmState, tmState \rangle$

$TS \triangleq \land pc[1] = \text{"TS"}$
         $\land \lor \land canCommit$
              $\land pc' = [pc \text{ EXCEPT } ![1] = \text{"TC"}]$
           $\lor \land canAbort$
              $\land pc' = [pc \text{ EXCEPT } ![1] = \text{"TA"}]$
         $\land \text{UNCHANGED } \langle rmState, tmState \rangle$

$TC \triangleq \land pc[1] = \text{"TC"}$
         $\land tmState' = \text{"committed"}$
         $\land pc' = [pc \text{ EXCEPT } ![1] = \text{"Done"}]$
         $\land \text{UNCHANGED } rmState$

$TA \triangleq \land pc[1] = \text{"TA"}$
         $\land tmState' = \text{"aborted"}$
         $\land pc' = [pc \text{ EXCEPT } ![1] = \text{"Done"}]$
         $\land \text{UNCHANGED } rmState$

$TManagerBackup \triangleq L1 \lor TS \lor TC \lor TA$

$Next \triangleq TManager \lor TManagerBackup$
           $\lor (\exists self \in RM : RManager(self))$
           $\lor$  Disjunct to prevent deadlock on termination
             $((\forall self \in ProcSet : pc[self] = \text{"Done"}) \land \text{UNCHANGED } vars)$

$Spec \triangleq \land Init \land \Box[Next]_{vars}$
          $\land \forall self \in RM : \text{WF}_{vars}(RManager(self))$
          $\land \text{WF}_{vars}(TManager)$
```

$\quad\quad \land \mathrm{WF}_{vars}(\mathit{TManagerBackup})$

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

END TRANSLATION

$\mathit{Termination2} \triangleq \Diamond(\forall \mathit{self} \in \mathit{ProcSet} : pc[\mathit{self}] = \text{"Done"})$
$\mathit{Consistency} \triangleq \forall rm1, rm2 \in RM : \neg(\mathit{rmState}[rm1] = \text{"aborted"} \land \mathit{rmState}[rm2] = \text{"committed"})$

\ ***ZHENG KAI* 50247576 **
\ ***XINBO YU* 50102922 **
\* Part 1.1
\* Model check *Consistency* and *Termination* with *RMMAYFAIL* = FALSE and *TMMAYFAIL* = FALSE,
\* it means that no *RM* fasle and no *TM* false, and there are no errors in my program.
\* Model check *Consistency* and *Termination* with *RMMAYFAIL* = TRUE and *TMMAYFAIL* = FALSE,
\* it means that *RM* can be fasle and no *TM* false, not all *RMs* can change their status to committed, and there are no errors in my program.

\* Part 1.2
\* Model check *Consistency* and *Termination* with *RMMAYFAIL* = FALSE and *TMMAYFAIL* = TRUE,
\* it means that no *RM* fasle and *TM* will be false, and the *Termination* property will be violated.
\* The error example is *rmState* < "aborted", "prepared", "aborted" > and *tmState* = *hidden*. When
\* *tmState* equals hidden, it doesn't make a decison. When the *RMs* read the state of the *TM*, it can't
\* return committed or aborted, so the *RMs* can't change their state.

\* Part 1.3
\* When set *RMMAYFAIL* TRUE and *TMMAYFAIL* TRUE, model check *Consistency* and *Termination*
\* Both of them are not violated. Because when *TM* fail, the *RMs* can connect to the Backup *TM*,
\* they can read the state of Backup *TM*, the Backup *TM* can reset the status of the tm, so the *RMs* can
\* change to their right final status according to *tmState*.

\* Modification History
\* Last modified *Tue Dec* 05 11:17:23 *EST* 2017 by *xinboyu*
\* Last modified *Tue Nov* 28 13:56:03 *EST* 2017 by kz-*pc*
\* Created Sun *Nov* 26 15:42:50 *EST* 2017 by *xinboyu*