Proof of Theorem: $TPSpec \Rightarrow TCSpec$

We prove that the Two-Phase Commit specification $TPSpec$ refines the Transaction Commit specification $TCSpec$, i.e.,

$$TPSpec \Rightarrow TCSpec$$

1. Initial State

The definition of $TPInit$:

$$rmState = [rm \in RM \mapsto working] \land ...$$

obviously implies $TCInit : rmState = [rm \in RM \mapsto working]$. Therefore,

$$TPInit \Rightarrow TCInit$$

2. Actions

We must show that every $TPNext$ action either updates $rmState$ in accordance with $TCNext$ or leaves it unchanged (i.e., a stuttering step for $TCSpec$), i.e,

$$TPNext \Rightarrow TCNext \lor (rmState = rmState')$$

Case 1: TMRcvPrepared(rm), TMCommit, TMAbort do not change $rmState$.

$$TMRcvPrepared(rm) \lor TMCommit \lor TMAbort \Rightarrow (rmState = rmState')$$

Case 2: RMPrepare(rm) is

$$\begin{aligned} &\land\, rmState[rm] = working \\ &\land\, rmState'[rm] = prepared \\ &\land\, ... \end{aligned}$$

Hence,

$$RMPrepare(rm) \Rightarrow Prepare(rm)$$

Case 3: RMChooseToAbort(rm) is

$$\begin{aligned} &\land\, rmState[rm] = working \\ &\land\, rmState'[rm] = aborted \\ &\land\, ... \end{aligned}$$

It matches the second case of $Decide(rm)$ in $TCNext$:

$$\begin{aligned}&\wedge\, rmState[rm] \in \{working, prepared\}\\&\wedge\, notCommitted\\&\wedge\, rmState'[rm] = aborted\end{aligned}$$

except, it is missing the condition

$$notCommitted = \forall rm \in RM : rmState[rm] \neq committed$$

i.e, no RM has committed yet. However,

$$TPInv1 : rmState[rm] = working \Rightarrow notCommitted$$

is an invariant of $TPNext$. This is because RMs commit only after TM sends *Commit* message. TM sends *Commit* message only after *every* RM is *prepared*, i.e., not *waiting*. Therefore,

$$RMChooseToAbort(rm) \Rightarrow \text{abort case of } Decide(rm)$$

Case 4: RMRcvAbortMsg(rm) is

$$\begin{aligned}&\wedge\, [type \mapsto Abort] \in msgs\\&\wedge\, rmState'[rm] = aborted\\&\wedge\, ...\end{aligned}$$

This should match the abort case of $Decide(rm)$, but note that we are missing $notCommitted$ and $rmState[rm] \in \{working, prepared\}$.

$$\begin{aligned}&\wedge\, rmState[rm] \in \{working, prepared\}\\&\wedge\, notCommitted\\&\wedge\, rmState'[rm] = aborted\end{aligned}$$

However,

$$TPInv2 : [type \mapsto Aborted] \in msgs \Rightarrow notCommitted$$

is an invariant of $TPSpec$. Why?

Now, recall that

$$notCommitted = \forall rm \in RM : rmState[rm] \neq committed$$

Hence, $rmState[rm]$ is either *aborted* in which case this is a stuttering step for $TCSpec$. Or $rmState[rm] \in \{working, prepared\}$ which matches the abort case of $Decide(rm)$. Hence,

$$RMRcvAbortMsg(rm) \Rightarrow (\text{abort case of } Decide(rm) \vee rmState = rmState')$$

Case 5: RMRcvCommitMsg(rm) similar to case 4,

$RMRcvCommitMsg(rm) \Rightarrow$ (commit case of $Decide(rm) \vee rmState = rmState'$)

□