

---

MODULE *Proposal*

---

EXTENDS *Configuration*, *Mastership*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

---

CONSTANT *NumProposals*

ASSUME *NumProposals*  $\in$  *Nat*

Transaction type constants

CONSTANTS

*ProposalChange*,  
*ProposalRollback*

Phase constants

CONSTANTS

*ProposalCommit*,  
*ProposalApply*

Status constants

CONSTANTS

*ProposalPending*,  
*ProposalInProgress*,  
*ProposalComplete*,  
*ProposalFailed*

CONSTANT *LogProposal*

ASSUME *LogProposal*  $\in$  BOOLEAN

A record of per-target proposals

VARIABLE *proposal*

---

LOCAL *CurrentState*  $\triangleq$  [  
     *proposals*       $\mapsto [i \in \{i \in \text{DOMAIN } proposal : proposal[i].state \neq Nil\} \mapsto proposal[i]]$ ,  
     *configuration*  $\mapsto configuration$ ,  
     *target*           $\mapsto target$ ,  
     *mastership*      $\mapsto mastership$ ,  
     *nodes*           $\mapsto node$ ]

LOCAL *SuccessorState*  $\triangleq$  [

$$\begin{array}{ll} \text{LOCAL } Log \stackrel{\Delta}{=} & \text{INSTANCE } Log \text{ WITH} \\ \text{File} & \leftarrow \text{"Proposal.log"}, \\ \text{CurrentState} & \leftarrow \text{CurrentState}, \\ \text{SuccessorState} & \leftarrow \text{SuccessorState} \\ \text{Enabled} & \leftarrow \text{LogProposal} \end{array}$$

2

$$\begin{aligned} & \begin{aligned} & ! [i].change.status &= ProposalPending, \\ & ! [i].rollback.revision &= rollbackRevision, \\ & ! [i].rollback.values &= rollbackValues \end{aligned} \\ \vee \wedge configuration' &= [configuration \text{ EXCEPT } !.commit.revision = i] \\ \wedge proposal' &= [proposal \text{ EXCEPT } ! [i].change.status = ProposalFailed] \\ \wedge \text{UNCHANGED } &\langle target \rangle \end{aligned}$$

$$ApplyChange(n, i) \triangleq$$

To apply a change, the apply index must be the prior index.

$$\wedge \text{configuration.apply.revision} = \text{configuration.apply.target}$$
$$\wedge proposal' = [proposal \text{ EXCEPT } ![i].change.status = ProposalInProgress]$$
$$\vee \wedge proposal[i].change.status = ProposalInProgress$$

Verify the node's connection to the target.

$$\wedge \text{mastership.conn} = \text{node}[n].\text{incarnation}$$
$$\wedge node[n].target = target.incarnation$$
$$\wedge \vee \wedge target' = [target \text{ EXCEPT } !.values = proposal[i].change.values @@ target.values]$$
$$\text{IN } \textit{configuration}' = [\textit{configuration} \text{ EXCEPT } \begin{array}{l} !.\textit{apply.revision} = i, \\ !.\textit{apply.incarnation} = \textit{target.incarnation}, \\ !.\textit{apply.values} = \textit{values} \end{array}]$$

If the proposal could not be applied, mark it failed but do not update the last applied index. The proposal must be rolled back before new proposals can be applied to the configuration/target.

 $\wedge$  UNCHANGED  $\langle configuration, target \rangle$



$$\begin{aligned}
& \wedge \text{ LET } \text{revision} \triangleq \text{proposal}[i].\text{rollback}.\text{revision} \\
& \quad \text{values} \triangleq \text{proposal}[i].\text{rollback}.\text{values} @@ \text{configuration}.\text{apply}.\text{values} \\
& \text{ IN } \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{apply}.\text{revision} = \text{revision}, \\
& \quad \quad \quad !.\text{apply}.\text{values} = \text{values}] \\
& \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{rollback}.\text{status} = \text{ProposalComplete}] \\
\text{ReconcileProposal}(n, i) & \triangleq \\
& \wedge \text{mastership}.\text{master} = n \\
& \wedge \vee \wedge \text{proposal}[i].\text{state} = \text{ProposalChange} \\
& \quad \wedge \vee \wedge \text{proposal}[i].\text{change}.\text{phase} = \text{ProposalCommit} \\
& \quad \quad \wedge \text{CommitChange}(n, i) \\
& \quad \vee \wedge \text{proposal}[i].\text{change}.\text{phase} = \text{ProposalApply} \\
& \quad \quad \wedge \text{ApplyChange}(n, i) \\
& \vee \wedge \text{proposal}[i].\text{state} = \text{ProposalRollback} \\
& \quad \wedge \vee \wedge \text{proposal}[i].\text{rollback}.\text{phase} = \text{ProposalCommit} \\
& \quad \quad \wedge \text{CommitRollback}(n, i) \\
& \quad \vee \wedge \text{proposal}[i].\text{rollback}.\text{phase} = \text{ProposalApply} \\
& \quad \quad \wedge \text{ApplyRollback}(n, i) \\
& \wedge \text{UNCHANGED } \langle \text{mastership}, \text{node} \rangle
\end{aligned}$$


---

Formal specification, constraints, and theorems.

$$\begin{aligned}
\text{InitProposal} & \triangleq \\
& \wedge \text{Log!Init} \\
& \wedge \text{proposal} = [ \\
& \quad i \in 1 \dots \text{NumProposals} \mapsto [ \\
& \quad \quad \text{state} \mapsto \text{Nil}, \\
& \quad \quad \text{change} \mapsto [ \\
& \quad \quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{Nil}]], \\
& \quad \quad \quad \text{phase} \mapsto \text{Nil}, \\
& \quad \quad \quad \text{status} \mapsto \text{Nil}], \\
& \quad \quad \text{rollback} \mapsto [ \\
& \quad \quad \quad \text{revision} \mapsto 0, \\
& \quad \quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{Nil}]], \\
& \quad \quad \quad \text{phase} \mapsto \text{Nil}, \\
& \quad \quad \quad \text{status} \mapsto \text{Nil}]] \\
& \quad ] \\
\text{NextProposal} & \triangleq \\
& \vee \exists n \in \text{Nodes} : \\
& \quad \exists i \in \text{DOMAIN } \text{proposal} : \\
& \quad \quad \text{Log!Action}(\text{ReconcileProposal}(n, i), [\text{node} \mapsto n, \text{index} \mapsto i])
\end{aligned}$$


---

\ \* Modification History  
\ \* Last modified *Fri Apr 21 19:15:11 PDT 2023* by *jhalterm*

\\* Last modified *Mon Feb 21 01:24:12 PST 2022* by *jordanhalterman*  
\\* Created Sun *Feb 20 10:07:16 PST 2022* by *jordanhalterman*