

---

MODULE *Proposals*

---

EXTENDS *Configurations, Southbound*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

---

**Transaction type constants**

CONSTANTS

*ProposalChange,*  
*ProposalRollback*

**Phase constants**

CONSTANTS

*ProposalInitialize,*  
*ProposalValidate,*  
*ProposalAbort,*  
*ProposalCommit,*  
*ProposalApply*

**Status constants**

CONSTANTS

*ProposalInProgress,*  
*ProposalComplete,*  
*ProposalFailed*

CONSTANTS

*ProposalValid,*  
*ProposalInvalid*

CONSTANTS

*ProposalSuccess,*  
*ProposalFailure*

**A record of per-target proposals**

VARIABLE *proposal*

---

LOCAL *InitState*  $\triangleq$

[*proposals*  $\mapsto$  *proposal*,  
*configurations*  $\mapsto$  *configuration*,  
*targets*  $\mapsto$  *target*,  
*masterships*  $\mapsto$  *mastership*]

$\text{LOCAL } \text{NextState} \triangleq$   
 $\quad [ \text{proposals} \mapsto \text{proposal}',$   
 $\quad \text{configurations} \mapsto \text{configuration}',$   
 $\quad \text{targets} \mapsto \text{target}',$   
 $\quad \text{masterships} \mapsto \text{mastership}' ]$   
 $\text{LOCAL } \text{Trace} \triangleq \text{INSTANCE } \text{Trace} \text{ WITH}$   
 $\quad \text{Module} \leftarrow \text{"Proposals"},$   
 $\quad \text{InitState} \leftarrow \text{InitState},$   
 $\quad \text{NextState} \leftarrow \text{NextState}$

---

**Reconcile a proposal**

$\text{ReconcileProposal}(n, t, i) \triangleq$   
 $\quad \wedge \vee \wedge \text{proposal}[t][i].\text{phase} = \text{ProposalInitialize}$   
 $\quad \wedge \text{proposal}[t][i].\text{state} = \text{ProposalInProgress}$   
 $\quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] =$   
 $\quad \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalComplete},$   
 $\quad \quad \quad ![i].\text{dependency.index} = \text{configuration}[t].\text{proposal.index}]]$   
 $\quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{proposal.index} = i]$   
 $\quad \wedge \text{UNCHANGED } \langle \text{target} \rangle$   
 $\quad \text{While in the } \text{Validate} \text{ phase, validate the proposed changes.}$   
 $\quad \text{If validation is successful, the proposal also records the changes}$   
 $\quad \text{required to roll back the proposal and the index to which to roll back.}$   
 $\quad \vee \wedge \text{proposal}[t][i].\text{phase} = \text{ProposalValidate}$   
 $\quad \wedge \text{proposal}[t][i].\text{state} = \text{ProposalInProgress}$   
 $\quad \wedge \text{configuration}[t].\text{commit.index} = \text{proposal}[t][i].\text{dependency.index}$   
 $\quad \text{For } \text{Change} \text{ proposals validate the set of requested changes.}$   
 $\quad \wedge \vee \wedge \text{proposal}[t][i].\text{type} = \text{ProposalChange}$   
 $\quad \quad \wedge \text{LET } \text{rollbackIndex} \triangleq \text{configuration}[t].\text{config.index}$   
 $\quad \quad \quad \text{rollbackValues} \triangleq [p \in \text{DOMAIN } \text{proposal}[t][i].\text{change.values} \mapsto$   
 $\quad \quad \quad \quad \text{IF } p \in \text{DOMAIN } \text{configuration}[t].\text{config.values} \text{ THEN}$   
 $\quad \quad \quad \quad \quad \text{configuration}[t].\text{config.values}[p]$   
 $\quad \quad \quad \quad \text{ELSE}$   
 $\quad \quad \quad \quad \quad [ \text{delete} \mapsto \text{TRUE} ]]$   
 $\quad \text{Model validation successes and failures with } \text{Valid} \text{ and } \text{Invalid} \text{ results.}$   
 $\quad \text{IN } \exists r \in \{ \text{ProposalValid}, \text{ProposalInvalid} \} :$   
 $\quad \quad \text{If the } \text{Change} \text{ is } \text{Valid}, \text{ record the changes required to roll}$   
 $\quad \quad \text{back the proposal and the index to which the rollback changes}$   
 $\quad \quad \text{will roll back the configuration.}$   
 $\quad \vee \wedge r = \text{ProposalValid}$   
 $\quad \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] =$   
 $\quad \quad \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{rollback} = [\text{index} \mapsto \text{rollbackIndex},$   
 $\quad \quad \quad \quad \text{values} \mapsto \text{rollbackValues}],$   
 $\quad \quad \quad \quad \quad ![i].\text{state} = \text{ProposalComplete}]$

$$\begin{aligned}
& \vee \wedge r = \text{ProposalInvalid} \\
& \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = \\
& \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalFailed}]] \\
& \text{For Rollback proposals, validate the rollback changes which are} \\
& \text{proposal being rolled back.} \\
& \vee \wedge \text{proposal}[t][i].\text{type} = \text{ProposalRollback} \\
& \quad \text{Rollbacks can only be performed on Change type proposals.} \\
& \wedge \vee \wedge \text{proposal}[t][\text{proposal}[t][i].\text{rollback.index}].\text{type} = \text{ProposalChange} \\
& \quad \text{Only roll back the change if it's the latest change made} \\
& \quad \text{to the configuration based on the configuration index.} \\
& \wedge \vee \wedge \text{configuration}[t].\text{config.index} = \text{proposal}[t][i].\text{rollback.index} \\
& \quad \wedge \text{LET } \text{changeIndex} \triangleq \text{proposal}[t][\text{proposal}[t][i].\text{rollback.index}].\text{rollback.index} \\
& \quad \quad \text{changeValues} \triangleq \text{proposal}[t][\text{proposal}[t][i].\text{rollback.index}].\text{rollback.values} \\
& \quad \quad \text{rollbackValues} \triangleq \text{proposal}[t][\text{proposal}[t][i].\text{rollback.index}].\text{change.values} \\
& \quad \text{IN } \exists r \in \{\text{ProposalValid}, \text{ProposalInvalid}\} : \\
& \quad \text{If the Rollback is Valid, record the changes required to} \\
& \quad \text{roll back the target proposal and the index to which the} \\
& \quad \text{configuration is being rolled back.} \\
& \vee \wedge r = \text{ProposalValid} \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = \\
& \quad \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{change} = [\text{index} \mapsto \text{changeIndex}, \\
& \quad \quad \quad \text{values} \mapsto \text{changeValues}], \\
& \quad \quad \quad ![i].\text{change} = [\text{index} \mapsto \text{proposal}[t][i].\text{change.index}, \\
& \quad \quad \quad \text{values} \mapsto \text{changeValues}], \\
& \quad \quad \quad ![i].\text{state} = \text{ProposalComplete}]] \\
& \vee \wedge r = \text{ProposalInvalid} \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = \\
& \quad \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalFailed}]] \\
& \quad \text{If the Rollback target is not the most recent change to the configuration,} \\
& \quad \text{fail validation for the proposal.} \\
& \vee \wedge \text{configuration}[t].\text{config.index} \neq \text{proposal}[t][i].\text{rollback.index} \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalFailed}]] \\
& \quad \text{If a Rollback proposal is attempting to roll back another Rollback,} \\
& \quad \text{fail validation for the proposal.} \\
& \vee \wedge \text{proposal}[t][\text{proposal}[t][i].\text{rollback.index}].\text{type} = \text{ProposalRollback} \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = \\
& \quad \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalFailed}]] \\
& \quad \wedge \text{UNCHANGED } \langle \text{configuration}, \text{target} \rangle \\
& \quad \text{While in the Commit state, commit the proposed changes to the configuration.} \\
& \vee \wedge \text{proposal}[t][i].\text{phase} = \text{ProposalCommit} \\
& \quad \wedge \text{proposal}[t][i].\text{state} = \text{ProposalInProgress} \\
& \quad \text{Only commit the proposal if the prior proposal has already been committed.} \\
& \quad \wedge \text{configuration}[t].\text{commit.index} = \text{proposal}[t][i].\text{dependency.index} \\
& \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{config.values} = \text{proposal}[t][i].\text{change.values}, \\
& \quad \quad \quad ![t].\text{config.index} = \text{proposal}[t][i].\text{change.index},
\end{aligned}$$

$\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalComplete}]]$   
 $\wedge \text{UNCHANGED } \langle \text{target} \rangle$   
 While in the *Apply* phase, apply the proposed changes to the target.  
 $\vee \wedge \text{proposal}[t][i].\text{phase} = \text{ProposalApply}$   
 $\wedge \text{proposal}[t][i].\text{state} = \text{ProposalInProgress}$   
 $\wedge \text{configuration}[t].\text{target.index} = \text{proposal}[t][i].\text{dependency.index}$   
 $\wedge \text{configuration}[t].\text{target.term} = \text{mastership}[t].\text{term}$   
 $\wedge \text{mastership}[t].\text{master} = n$   
 Model successful and failed target update requests.  
 $\wedge \exists r \in \{\text{ProposalSuccess}, \text{ProposalFailure}\} :$   
 $\vee \wedge r = \text{ProposalSuccess}$   
 $\wedge \text{target}' = [\text{target} \text{ EXCEPT } ![t] = \text{proposal}[t][i].\text{change.values} @@ \text{target}[t]]$   
 $\wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ! [t].\text{target.index} = i,$   
 $\quad ! [t].\text{target.values} = \text{proposal}[t][i].\text{change.values}$   
 $\quad \quad @@ \text{configuration}[t].\text{target.values}]$   
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalComplete}]]$   
 If the proposal could not be applied, update the configuration's applied index and mark the proposal *Failed*.  
 $\vee \wedge r = \text{ProposalFailure}$   
 $\wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{target.index} = i]$   
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalFailed}]]$   
 $\wedge \text{UNCHANGED } \langle \text{target} \rangle$   
 $\vee \wedge \text{proposal}[t][i].\text{phase} = \text{ProposalAbort}$   
 $\wedge \text{proposal}[t][i].\text{state} = \text{ProposalInProgress}$   
 The *commit.index* will always be greater than or equal to the *target.index*.  
 If only the *commit.index* matches the proposal's *dependency.index*, update the *commit.index* to enable commits of later proposals, but do not mark the *Abort* phase *Complete* until the *target.index* has been incremented.  
 $\wedge \vee \wedge \text{configuration}[t].\text{commit.index} = \text{proposal}[t][i].\text{dependency.index}$   
 $\wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{commit.index} = i]$   
 $\wedge \text{UNCHANGED } \langle \text{proposal} \rangle$   
 If the configuration's *target.index* matches the proposal's *dependency.index*, update the *target.index* and mark the proposal *Complete* for the *Abort* phase.  
 $\vee \wedge \text{configuration}[t].\text{commit.index} \geq i$   
 $\wedge \text{configuration}[t].\text{target.index} = \text{proposal}[t][i].\text{dependency.index}$   
 $\wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{target.index} = i]$   
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\text{proposal}[t] \text{ EXCEPT } ![i].\text{state} = \text{ProposalComplete}]]$   
 If both the configuration's *commit.index* and *target.index* match the proposal's *dependency.index*, update the *commit.index* and *target.index* and mark the proposal *Complete* for the *Abort* phase.  
 $\vee \wedge \text{configuration}[t].\text{commit.index} = \text{proposal}[t][i].\text{dependency.index}$   
 $\wedge \text{configuration}[t].\text{target.index} = \text{proposal}[t][i].\text{dependency.index}$   
 $\wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{commit.index} = i,$

