
MODULE *Proposal*

EXTENDS *Configuration, 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*]

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

Reconcile a proposal

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

$$\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}$$

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

$$\begin{aligned} & \text{!}[t].target.index = i] \\ \wedge \text{proposal}' &= [\text{proposal} \text{ EXCEPT !}[t] = [\text{proposal}[t] \text{ EXCEPT !}[i].state = \text{ProposalComplete}]] \\ \wedge \text{UNCHANGED } &\langle target \rangle \\ \wedge \text{UNCHANGED } &\langle mastership \rangle \end{aligned}$$

Formal specification, constraints, and theorems.

$$\begin{aligned} InitProposal &\triangleq \\ \wedge proposal &= [t \in \text{DOMAIN } Target \mapsto \\ &\quad [i \in \{ \} \mapsto \\ &\quad \quad [phase \mapsto ProposalInitialize, \\ &\quad \quad \quad state \mapsto ProposalInProgress]]] \\ \wedge Trace!Init & \end{aligned}$$
$$\begin{aligned} \text{NextProposal} &\triangleq \\ &\vee \exists n \in \text{Node} : \\ &\quad \exists t \in \text{DOMAIN } \text{proposal} : \\ &\quad \exists i \in \text{DOMAIN } \text{proposal}[t] : \\ &\quad \text{Trace!Step}(\text{"Reconcile"}, \text{ReconcileProposal}(n, t, i), [\text{node} \mapsto n, \text{target} \mapsto t, \text{index} \mapsto i]) \end{aligned}$$

```
\* Modification History
\* Last modified Sun Feb 20 09:01:20 PST 2022 by jordanhalterman
\* Created Sun Feb 20 02:20:56 PST 2022 by jordanhalterman
```