
MODULE *Config*

INSTANCE *Naturals*
 INSTANCE *FiniteSets*
 INSTANCE *Sequences*
 LOCAL INSTANCE *TLC*

This section specifies constant parameters for the model.
 CONSTANT *None*
 ASSUME *None* ∈ STRING
 CONSTANT *Node*
 ASSUME $\forall n \in \text{Node} : n \in \text{STRING}$
 CONSTANTS
 Change,
 Rollback
Event $\triangleq \{ \text{Change}, \text{Rollback} \}$
 ASSUME $\forall e \in \text{Event} : e \in \text{STRING}$
 CONSTANTS
 Commit,
 Apply
Phase $\triangleq \{ \text{Commit}, \text{Apply} \}$
 ASSUME $\forall p \in \text{Phase} : p \in \text{STRING}$
 CONSTANTS
 Pending,
 InProgress,
 Complete,
 Aborted,
 Failed
State $\triangleq \{ \text{Pending}, \text{InProgress}, \text{Complete}, \text{Aborted}, \text{Failed} \}$
Working $\triangleq \{ \text{Pending}, \text{InProgress} \}$
Finished $\triangleq \{ \text{Complete}, \text{Aborted}, \text{Failed} \}$
 ASSUME $\forall s \in \text{State} : s \in \text{STRING}$

CONSTANT *Path*
 ASSUME $\forall p \in Path : p \in \text{STRING}$
 CONSTANT *Value*
 ASSUME $\forall v \in Value : v \in \text{STRING}$
 $AllValues \triangleq Value \cup \{None\}$
 CONSTANT *NumProposals*
 ASSUME $NumProposals \in Nat$

This section defines model state variables.

$proposal \triangleq [i \in 1 \dots Nat \mapsto [$
 $phase \mapsto Phase,$
 $change \mapsto [$
 $values \mapsto Change,$
 $commit \mapsto State,$
 $apply \mapsto State],$
 $rollback \mapsto [$
 $index \mapsto Nat,$
 $values \mapsto Change,$
 $commit \mapsto State,$
 $apply \mapsto State]]]$
 $configuration \triangleq [$
 $committed \mapsto [$
 $index \mapsto Nat,$
 $values \mapsto Change],$
 $applied \mapsto [$
 $index \mapsto Nat,$
 $values \mapsto Change,$
 $term \mapsto Nat]]]$
 $mastership \triangleq [$
 $master \mapsto \text{STRING},$
 $term \mapsto Nat,$
 $conn \mapsto Nat]$
 $conn \triangleq [n \in Node \mapsto [$
 $id \mapsto Nat,$
 $connected \mapsto \text{BOOLEAN}]]$
 $target \triangleq [$
 $id \mapsto Nat,$
 $values \mapsto Change,$
 $running \mapsto \text{BOOLEAN}]$
 VARIABLE *proposal*

VARIABLE *configuration*

VARIABLE *mastership*

VARIABLE *conn*

VARIABLE *target*

VARIABLE *history*

$\text{vars} \triangleq \langle \text{proposal}, \text{configuration}, \text{mastership}, \text{conn}, \text{target}, \text{history} \rangle$

This section models configuration target.

$\text{StartTarget} \triangleq$

$\wedge \neg \text{target.running}$
 $\wedge \text{target}' = [\text{target} \text{ EXCEPT } !.id = \text{target.id} + 1,$
 $\quad \quad \quad !.running = \text{TRUE}]$
 $\wedge \text{UNCHANGED } \langle \text{proposal}, \text{configuration}, \text{mastership}, \text{conn}, \text{history} \rangle$

$\text{StopTarget} \triangleq$

$\wedge \text{target.running}$
 $\wedge \text{target}' = [\text{target} \text{ EXCEPT } !.running = \text{FALSE},$
 $\quad \quad \quad !.values = [p \in \{\} \mapsto [value \mapsto \text{None}]]]$
 $\wedge \text{conn}' = [n \in \text{Node} \mapsto [\text{conn}[n] \text{ EXCEPT } !.connected = \text{FALSE}]]$
 $\wedge \text{UNCHANGED } \langle \text{proposal}, \text{configuration}, \text{mastership}, \text{history} \rangle$

This section models nodes connection to the configuration target.

$\text{ConnectNode}(n) \triangleq$

$\wedge \neg \text{conn}[n].connected$
 $\wedge \text{target.running}$
 $\wedge \text{conn}' = [\text{conn} \text{ EXCEPT } ![n].id = \text{conn}[n].id + 1,$
 $\quad \quad \quad ![n].connected = \text{TRUE}]$
 $\wedge \text{UNCHANGED } \langle \text{proposal}, \text{configuration}, \text{mastership}, \text{target}, \text{history} \rangle$

$\text{DisconnectNode}(n) \triangleq$

$\wedge \text{conn}[n].connected$
 $\wedge \text{conn}' = [\text{conn} \text{ EXCEPT } ![n].connected = \text{FALSE}]$
 $\wedge \text{UNCHANGED } \langle \text{proposal}, \text{configuration}, \text{mastership}, \text{target}, \text{history} \rangle$

This section models *mastership* reconciliation.

$\text{ReconcileMastership}(n) \triangleq$

$\wedge \vee \wedge \text{conn}[n].connected$
 $\quad \wedge \text{mastership.master} = \text{None}$

$$\begin{aligned}
& \text{IN} \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{committed.values} = \text{values}] \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{change.commit} = \text{Complete}] \\
& \quad \wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Change}, \text{phase} \mapsto \text{Commit}, \text{index} \mapsto i]) \\
& \wedge \text{UNCHANGED} \langle \text{mastership}, \text{conn}, \text{target} \rangle \\
\text{ApplyChange}(n, i) & \triangleq \\
& \wedge \vee \wedge \text{proposal}[i].\text{change.apply} = \text{Pending} \\
& \quad \wedge \text{proposal}[i].\text{rollback.apply} = \text{None} \\
& \wedge \vee \wedge \text{proposal}[i].\text{change.commit} = \text{Complete} \\
& \quad \wedge \forall j \in \text{DOMAIN } \text{proposal} : j < i \Rightarrow \\
& \quad \quad \vee \wedge \text{proposal}[j].\text{change.apply} = \text{Complete} \\
& \quad \quad \wedge \text{proposal}[j].\text{rollback.apply} \notin \text{Working} \\
& \quad \quad \vee \wedge \text{proposal}[j].\text{change.apply} = \text{Failed} \\
& \quad \quad \wedge \text{proposal}[j].\text{rollback.apply} = \text{Complete} \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{change.apply} = \text{InProgress}] \\
& \quad \vee \wedge \text{proposal}[i].\text{change.commit} \in \{\text{Aborted}, \text{Failed}\} \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{change.apply} = \text{Aborted}] \\
& \wedge \text{UNCHANGED} \langle \text{configuration}, \text{target}, \text{history} \rangle \\
& \vee \wedge \text{proposal}[i].\text{change.apply} = \text{InProgress} \\
& \quad \text{Verify the applied term is the current } \text{mastership} \text{ term to ensure the} \\
& \quad \text{configuration has been synchronized following restarts.} \\
& \quad \wedge \text{configuration.applied.term} = \text{mastership.term} \\
& \quad \text{Verify the node's connection to the target.} \\
& \quad \wedge \text{conn}[n].\text{connected} \\
& \quad \wedge \text{mastership.conn} = \text{conn}[n].\text{id} \\
& \quad \wedge \text{target.running} \\
& \quad \text{Model successful and failed target update requests.} \\
& \wedge \vee \wedge \text{LET } \text{values} \triangleq [p \in \text{DOMAIN } \text{proposal}[i].\text{values} \mapsto \\
& \quad \quad \quad [\text{index} \mapsto i, \text{value} \mapsto \text{proposal}[i].\text{values}[p]]] \\
& \quad \text{IN} \quad \wedge \text{target}' = [\text{target} \text{ EXCEPT } !.\text{values} = \text{values} @@ \text{target.values}] \\
& \quad \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{applied.values} = \text{values} @@ \\
& \quad \quad \quad \text{configuration.applied.values}] \\
& \quad \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{change.apply} = \text{Complete}] \\
& \quad \quad \wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Change}, \text{phase} \mapsto \text{Apply}, \text{index} \mapsto i]) \\
& \quad \vee \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{change.apply} = \text{Failed}] \\
& \quad \wedge \text{UNCHANGED} \langle \text{configuration}, \text{target}, \text{history} \rangle \\
& \wedge \text{UNCHANGED} \langle \text{mastership}, \text{conn} \rangle \\
\text{CommitRollback}(n, i) & \triangleq \\
& \wedge \vee \wedge \text{proposal}[i].\text{rollback.commit} = \text{Pending} \\
& \quad \wedge \forall j \in \text{DOMAIN } \text{proposal} : j > i \wedge \text{proposal}[j].\text{phase} \neq \text{None} \Rightarrow \\
& \quad \quad \text{proposal}[j].\text{rollback.commit} \in \text{Finished} \\
& \wedge \vee \wedge \text{proposal}[i].\text{change.commit} = \text{Pending} \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{change.commit} = \text{Aborted}, \\
& \quad \quad \quad ![i].\text{rollback.commit} = \text{Complete}]
\end{aligned}$$

$$\begin{aligned}
& \wedge \text{proposal}[j].\text{change.apply} = \text{Complete} \\
& \wedge \text{proposal}[j].\text{rollback.apply} \neq \text{Complete} \} \\
\text{paths} & \triangleq \{p \in \text{DOMAIN configuration.applied.values} : \\
& \quad \exists j \in \text{changes} : p \in \text{DOMAIN proposal}[j].\text{values} \} \\
\text{indexes} & \triangleq [p \in \text{paths} \mapsto \text{CHOOSE } j \in \text{changes} : \\
& \quad \wedge p \in \text{DOMAIN proposal}[j].\text{values} \\
& \quad \wedge \neg \exists k \in \text{changes} : k > j \wedge p \in \text{DOMAIN proposal}[k].\text{values}] \\
\text{values} & \triangleq [p \in \text{DOMAIN configuration.applied.values} \mapsto \\
& \quad \text{IF } p \in \text{paths} \text{ THEN} \\
& \quad \quad [index \mapsto \text{indexes}[p], \text{value} \mapsto \text{proposal}[\text{indexes}[p]].\text{values}[p]] \\
& \quad \text{ELSE} \\
& \quad \quad [index \mapsto 0, \text{value} \mapsto \text{None}]] \\
& \text{IN} \\
& \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{applied.values} = \text{values}] \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{rollback.apply} = \text{Complete}] \\
& \quad \wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Rollback}, \text{phase} \mapsto \text{Apply}, \text{index} \mapsto i]) \\
& \wedge \text{UNCHANGED } \langle \text{mastership}, \text{conn} \rangle \\
\text{ReconcileProposal}(n, i) & \triangleq \\
& \quad \wedge \text{mastership.master} = n \\
& \quad \wedge \vee \text{CommitChange}(n, i) \\
& \quad \quad \vee \text{ApplyChange}(n, i) \\
& \quad \quad \vee \text{CommitRollback}(n, i) \\
& \quad \quad \vee \text{ApplyRollback}(n, i) \\
& \quad \wedge \text{UNCHANGED } \langle \text{mastership}, \text{conn} \rangle
\end{aligned}$$

This section models changes to the proposal queue.

Rollback proposed change at index 'i'

$\wedge \text{UNCHANGED } \langle \text{configuration}, \text{mastership}, \text{conn}, \text{target}, \text{history} \rangle$

Formal specification, constraints, and theorems.

$\text{Init} \triangleq$

$$\begin{aligned} &\wedge \text{proposal} = [\\ &\quad i \in 1 \dots \text{NumProposals} \mapsto [\\ &\quad \quad \text{phase} \mapsto \text{None}, \\ &\quad \quad \text{values} \mapsto [p \in \{\} \mapsto \text{None}], \\ &\quad \quad \text{change} \mapsto [\\ &\quad \quad \quad \text{commit} \mapsto \text{None}, \\ &\quad \quad \quad \text{apply} \mapsto \text{None}], \\ &\quad \quad \text{rollback} \mapsto [\\ &\quad \quad \quad \text{commit} \mapsto \text{None}, \\ &\quad \quad \quad \text{apply} \mapsto \text{None}]]] \\ &\wedge \text{configuration} = [\\ &\quad \text{committed} \mapsto [\\ &\quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]]], \\ &\quad \text{applied} \mapsto [\\ &\quad \quad \text{term} \mapsto 0, \\ &\quad \quad \text{target} \mapsto 0, \\ &\quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]]], \\ &\quad \text{status} \mapsto \text{Pending}] \\ &\wedge \text{mastership} = [\text{master} \mapsto \text{None}, \text{term} \mapsto 0, \text{conn} \mapsto 0] \\ &\wedge \text{conn} = [n \in \text{Node} \mapsto [\text{id} \mapsto 0, \text{connected} \mapsto \text{FALSE}]] \\ &\wedge \text{target} = [\\ &\quad \text{id} \mapsto 0, \\ &\quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\ &\quad \text{running} \mapsto \text{FALSE}] \\ &\wedge \text{history} = \langle \rangle \end{aligned}$$

$\text{Next} \triangleq$

$$\begin{aligned} &\vee \exists i \in 1 \dots \text{NumProposals} : \\ &\quad \vee \text{ProposeChange}(i) \\ &\quad \vee \text{ProposeRollback}(i) \\ &\vee \exists n \in \text{Node}, i \in \text{DOMAIN } \text{proposal} : \text{ReconcileProposal}(n, i) \\ &\vee \exists n \in \text{Node} : \text{ReconcileConfiguration}(n) \\ &\vee \exists n \in \text{Node} : \text{ReconcileMastership}(n) \\ &\vee \exists n \in \text{Node} : \\ &\quad \vee \text{ConnectNode}(n) \\ &\quad \vee \text{DisconnectNode}(n) \\ &\vee \text{StartTarget} \\ &\vee \text{StopTarget} \end{aligned}$$

$\text{Spec} \triangleq$

$$\begin{aligned}
& \wedge \text{Init} \\
& \wedge \square[Next]_{vars} \\
& \wedge \forall i \in 1 \dots NumProposals : WF_{vars}(ProposeChange(i) \vee ProposeRollback(i)) \\
& \wedge \forall n \in Node, i \in 1 \dots NumProposals : WF_{vars}(ReconcileProposal(n, i)) \\
& \wedge \forall n \in Node : WF_{\langle configuration, mastership, conn, target \rangle}(ReconcileConfiguration(n)) \\
& \wedge \forall n \in Node : WF_{\langle mastership, conn, target \rangle}(ReconcileMastership(n)) \\
& \wedge \forall n \in Node : WF_{\langle conn, target \rangle}(ConnectNode(n) \vee DisconnectNode(n)) \\
& \wedge WF_{\langle target \rangle}(StartTarget) \\
& \wedge WF_{\langle target \rangle}(StopTarget) \\
\\
IsOrderedChange(p, i) & \triangleq \\
& \wedge history[i].type = Change \\
& \wedge history[i].phase = p \\
& \wedge \neg \exists j \in \text{DOMAIN } history : \\
& \quad \wedge j < i \\
& \quad \wedge history[j].type = Change \\
& \quad \wedge history[j].phase = p \\
& \quad \wedge history[j].index \geq history[i].index \\
\\
IsOrderedRollback(p, i) & \triangleq \\
& \wedge history[i].type = Rollback \\
& \wedge history[i].phase = p \\
& \wedge \neg \exists j \in \text{DOMAIN } history : \\
& \quad \wedge j < i \\
& \quad \wedge history[j].type = Change \\
& \quad \wedge history[j].phase = p \\
& \quad \wedge history[j].index > history[i].index \\
& \wedge \neg \exists k \in \text{DOMAIN } history : \\
& \quad \wedge k > j \\
& \quad \wedge k < i \\
& \quad \wedge history[k].type = Rollback \\
& \quad \wedge history[k].phase = p \\
& \quad \wedge history[k].index = history[j].index \\
\\
Order & \triangleq \\
& \wedge \forall i \in \text{DOMAIN } history : \\
& \quad \vee IsOrderedChange(Commit, i) \\
& \quad \vee IsOrderedChange(Apply, i) \\
& \quad \vee IsOrderedRollback(Commit, i) \\
& \quad \vee IsOrderedRollback(Apply, i) \\
& \wedge \forall i \in \text{DOMAIN } proposal : \\
& \quad \wedge proposal[i].change.apply = Failed \\
& \quad \wedge proposal[i].rollback.apply \neq Complete \\
& \Rightarrow \forall j \in \text{DOMAIN } proposal : j > i \Rightarrow \\
& \quad proposal[j].change.apply \in \{None, Pending, Aborted\}
\end{aligned}$$

$$\begin{aligned}
\text{AdditiveChanges} &\triangleq \\
&\wedge \forall i \in \text{DOMAIN } \text{proposal} : \\
&\quad \wedge \text{proposal}[i].\text{change.commit} = \text{Pending} \\
&\quad \wedge \text{proposal}'[i].\text{change.commit} = \text{InProgress} \\
&\quad \Rightarrow \forall j \in \text{DOMAIN } \text{proposal} : j < i \Rightarrow \\
&\quad \quad \wedge \text{proposal}[j].\text{phase} = \text{Change} \Rightarrow \text{proposal}[j].\text{change.commit} \in \text{Finished} \\
&\quad \quad \wedge \text{proposal}[j].\text{phase} = \text{Rollback} \Rightarrow \text{proposal}[j].\text{rollback.commit} \in \text{Finished} \\
&\wedge \forall i \in \text{DOMAIN } \text{proposal} : \\
&\quad \wedge \text{proposal}[i].\text{change.apply} = \text{Pending} \\
&\quad \wedge \text{proposal}'[i].\text{change.apply} = \text{InProgress} \\
&\quad \Rightarrow \forall j \in \text{DOMAIN } \text{proposal} : j < i \Rightarrow \\
&\quad \quad \wedge \text{proposal}[j].\text{phase} = \text{Change} \Rightarrow \text{proposal}[j].\text{change.apply} \in \text{Finished} \\
&\quad \quad \wedge \text{proposal}[j].\text{phase} = \text{Rollback} \Rightarrow \text{proposal}[j].\text{rollback.apply} \in \text{Finished}
\end{aligned}$$

$$\begin{aligned}
\text{SubtractiveRollbacks} &\triangleq \\
&\wedge \forall i \in \text{DOMAIN } \text{proposal} : \\
&\quad \wedge \text{proposal}[i].\text{rollback.commit} = \text{Pending} \\
&\quad \wedge \text{proposal}'[i].\text{rollback.commit} = \text{InProgress} \\
&\quad \Rightarrow \forall j \in \text{DOMAIN } \text{proposal} : j > i \wedge \text{proposal}[j].\text{phase} \neq \text{None} \Rightarrow \\
&\quad \quad \text{proposal}[j].\text{rollback.commit} = \text{Complete} \\
&\wedge \forall i \in \text{DOMAIN } \text{proposal} : \\
&\quad \wedge \text{proposal}[i].\text{rollback.apply} = \text{Pending} \\
&\quad \wedge \text{proposal}'[i].\text{rollback.apply} = \text{InProgress} \\
&\quad \Rightarrow \forall j \in \text{DOMAIN } \text{proposal} : j > i \wedge \text{proposal}[j].\text{phase} \neq \text{None} \Rightarrow \\
&\quad \quad \text{proposal}[j].\text{rollback.apply} = \text{Complete}
\end{aligned}$$

$$\text{Sequential} \triangleq \Box[\text{AdditiveChanges} \wedge \text{SubtractiveRollbacks}]_{\langle \text{proposal} \rangle}$$

$$\begin{aligned}
\text{Consistency} &\triangleq \\
&\wedge \text{target.running} \\
&\wedge \text{configuration.status} = \text{Complete} \\
&\wedge \text{configuration.applied.target} = \text{target.id} \\
&\Rightarrow \forall i \in \text{DOMAIN } \text{proposal} : \\
&\quad \wedge \text{proposal}[i].\text{change.apply} = \text{Complete} \\
&\quad \wedge \text{proposal}[i].\text{rollback.apply} \neq \text{Complete} \\
&\quad \Rightarrow \forall p \in \text{DOMAIN } \text{proposal}[i].\text{values} : \\
&\quad \quad \wedge \neg \exists j \in \text{DOMAIN } \text{proposal} : \\
&\quad \quad \quad \wedge j > i \\
&\quad \quad \quad \wedge \text{proposal}[j].\text{change.apply} = \text{Complete} \\
&\quad \quad \quad \wedge \text{proposal}[j].\text{rollback.apply} \neq \text{Complete} \\
&\quad \Rightarrow \wedge p \in \text{DOMAIN } \text{target.values} \\
&\quad \quad \wedge \text{target.values}[p].\text{value} = \text{proposal}[i].\text{values}[p].\text{value} \\
&\quad \quad \wedge \text{target.values}[p].\text{index} = i
\end{aligned}$$

$$\text{Safety} \triangleq \Box(\text{Order} \wedge \text{Consistency})$$

THEOREM $Spec \Rightarrow Safety$

$Termination \triangleq$

$$\begin{aligned}
& \forall i \in 1 \dots NumProposals : \\
& \quad \wedge proposal[i].change.commit = Pending \leadsto \\
& \quad \quad proposal[i].change.commit \in Finished \\
& \quad \wedge proposal[i].change.apply = Pending \leadsto \\
& \quad \quad proposal[i].change.apply \in Finished \\
& \quad \wedge proposal[i].rollback.commit = Pending \leadsto \\
& \quad \quad proposal[i].rollback.commit \in Finished \\
& \quad \wedge proposal[i].rollback.apply = Pending \leadsto \\
& \quad \quad proposal[i].rollback.apply \in Finished
\end{aligned}$$

$Liveness \triangleq Termination$

THEOREM $Spec \Rightarrow Liveness$
