
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{ELSE} \\
& \quad [index \mapsto 0, value \mapsto \text{None}] \\
& \text{committedValues} \triangleq [p \in \text{DOMAIN } proposal[i].change.values \mapsto \\
& \quad proposal[i].change.values[p] @@ [index \mapsto i]] @@ \\
& \quad configuration.committed.values \\
\text{IN} \quad & \wedge configuration' = [configuration \text{ EXCEPT } !.committed.index = i, \\
& \quad !.committed.values = committedValues] \\
& \wedge proposal' = [proposal \text{ EXCEPT } ![i].rollback.index = rollbackIndex, \\
& \quad ![i].rollback.values = rollbackValues, \\
& \quad ![i].change.commit = Complete] \\
& \wedge history' = Append(history, [type \mapsto Change, phase \mapsto Commit, index \mapsto i]) \\
& \vee \wedge proposal' = [proposal \text{ EXCEPT } ![i].change.commit = Failed] \\
& \wedge \text{UNCHANGED } \langle configuration, history \rangle \\
& \wedge \text{UNCHANGED } \langle mastership, conn, target \rangle \\
\text{ApplyChange}(n, i) \triangleq \\
& \wedge \vee \wedge proposal[i].change.apply = Pending \\
& \quad \wedge proposal[i].rollback.apply = None \\
& \wedge \vee \wedge proposal[i].change.commit = Complete \\
& \quad \wedge \forall j \in \text{DOMAIN } proposal : j < i \Rightarrow \\
& \quad \vee \wedge proposal[j].change.apply = Complete \\
& \quad \quad \wedge proposal[j].rollback.apply \notin \text{Working} \\
& \quad \vee \wedge proposal[j].change.apply = Failed \\
& \quad \quad \wedge proposal[j].rollback.apply = Complete \\
& \quad \wedge proposal' = [proposal \text{ EXCEPT } ![i].change.apply = InProgress] \\
& \vee \wedge proposal[i].change.commit \in \{Aborted, Failed\} \\
& \quad \wedge proposal' = [proposal \text{ EXCEPT } ![i].change.apply = Aborted] \\
& \wedge \text{UNCHANGED } \langle configuration, target, history \rangle \\
& \vee \wedge proposal[i].change.apply = InProgress \\
& \quad \text{Verify the applied term is the current } mastership \text{ term to ensure the} \\
& \quad \text{configuration has been synchronized following restarts.} \\
& \quad \wedge configuration.applied.term = mastership.term \\
& \quad \text{Verify the node's connection to the target.} \\
& \quad \wedge conn[n].connected \\
& \quad \wedge mastership.conn = conn[n].id \\
& \quad \wedge target.running \\
& \quad \text{Model successful and failed target update requests.} \\
& \wedge \vee \wedge \text{LET } values \triangleq [p \in \text{DOMAIN } proposal[i].change.values \mapsto \\
& \quad proposal[i].change.values[p] @@ [index \mapsto i]] \\
\text{IN} \quad & \wedge target' = [target \text{ EXCEPT } !.values = values @@ target.values] \\
& \quad \wedge configuration' = [configuration \text{ EXCEPT } !.applied.index = i, \\
& \quad \quad !.applied.values = values @@ \\
& \quad \quad configuration.applied.values] \\
& \quad \wedge proposal' = [proposal \text{ EXCEPT } ![i].change.apply = Complete] \\
& \quad \wedge history' = Append(history, [type \mapsto Change, phase \mapsto Apply, index \mapsto i])
\end{aligned}$$

[illegible]

$$\begin{aligned}
& \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{applied.index} = \text{index}, \\
& \quad \quad \quad !.\text{applied.values} = \text{values}] \\
& \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{rollback.apply} = \text{Complete}] \\
& \wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Rollback}, \text{phase} \mapsto \text{Apply}, \text{index} \mapsto i]) \\
& \wedge \text{UNCHANGED } \langle \text{mastership}, \text{conn} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{ReconcileProposal}(n, i) & \triangleq \\
& \wedge \text{mastership.master} = n \\
& \wedge \vee \text{CommitChange}(n, i) \\
& \quad \vee \text{ApplyChange}(n, i) \\
& \quad \vee \text{CommitRollback}(n, i) \\
& \quad \vee \text{ApplyRollback}(n, i) \\
& \wedge \text{UNCHANGED } \langle \text{mastership}, \text{conn} \rangle
\end{aligned}$$

This section models changes to the proposal queue.

$$\begin{aligned}
& \text{Propose change at index 'i'} \\
\text{ProposeChange}(i) & \triangleq \\
& \wedge \text{proposal}[i].\text{phase} = \text{None} \\
& \wedge i - 1 \in \text{DOMAIN } \text{proposal} \Rightarrow \text{proposal}[i - 1].\text{phase} \neq \text{None} \\
& \wedge \exists p \in \text{Path}, v \in \text{AllValues} : \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{phase} = \text{Change}, \\
& \quad \quad \quad ![i].\text{change.values} = (p :> [value \mapsto v]), \\
& \quad \quad \quad ![i].\text{change.commit} = \text{Pending}, \\
& \quad \quad \quad ![i].\text{change.apply} = \text{Pending}] \\
& \wedge \text{UNCHANGED } \langle \text{configuration}, \text{mastership}, \text{conn}, \text{target}, \text{history} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{Rollback proposed change at index 'i'} \\
\text{ProposeRollback}(i) & \triangleq \\
& \wedge \text{proposal}[i].\text{phase} = \text{Change} \\
& \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{phase} = \text{Rollback}, \\
& \quad \quad \quad ![i].\text{rollback.commit} = \text{Pending}, \\
& \quad \quad \quad ![i].\text{rollback.apply} = \text{Pending}] \\
& \wedge \text{UNCHANGED } \langle \text{configuration}, \text{mastership}, \text{conn}, \text{target}, \text{history} \rangle
\end{aligned}$$

Formal specification, constraints, and theorems.

$$\begin{aligned}
\text{Init} & \triangleq \\
& \wedge \text{proposal} = [\\
& \quad i \in 1 \dots \text{NumProposals} \mapsto [\\
& \quad \quad \text{phase} \mapsto \text{None}, \\
& \quad \quad \text{change} \mapsto [\\
& \quad \quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\
& \quad \quad \quad \text{commit} \mapsto \text{None}, \\
& \quad \quad] \\
& \quad]
\end{aligned}$$

$$\begin{aligned}
& \text{apply} \mapsto \text{None}], \\
& \text{rollback} \mapsto [\\
& \quad \text{index} \mapsto 0, \\
& \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\
& \quad \text{commit} \mapsto \text{None}, \\
& \quad \text{apply} \mapsto \text{None}]]] \\
\wedge \text{configuration} = [\\
& \quad \text{committed} \mapsto [\\
& \quad \quad \text{index} \mapsto 0, \\
& \quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\
& \quad \text{applied} \mapsto [\\
& \quad \quad \text{index} \mapsto 0, \\
& \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 \\
\text{Next} \triangleq \\
& \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} \\
\text{Spec} \triangleq \\
& \wedge \text{Init} \\
& \wedge \Box [\text{Next}]_{\text{vars}} \\
& \wedge \forall i \in 1 \dots \text{NumProposals} : \text{WF}_{\text{vars}}(\text{ProposeChange}(i) \vee \text{ProposeRollback}(i)) \\
& \wedge \forall n \in \text{Node}, i \in 1 \dots \text{NumProposals} : \text{WF}_{\text{vars}}(\text{ReconcileProposal}(n, i)) \\
& \wedge \forall n \in \text{Node} : \text{WF}_{\langle \text{configuration}, \text{mastership}, \text{conn}, \text{target} \rangle}(\text{ReconcileConfiguration}(n)) \\
& \wedge \forall n \in \text{Node} : \text{WF}_{\langle \text{mastership}, \text{conn}, \text{target} \rangle}(\text{ReconcileMastership}(n)) \\
& \wedge \forall n \in \text{Node} : \text{WF}_{\langle \text{conn}, \text{target} \rangle}(\text{ConnectNode}(n) \vee \text{DisconnectNode}(n))
\end{aligned}$$

$$\begin{aligned}
& \wedge \text{WF}_{\langle \text{target} \rangle}(\text{StartTarget}) \\
& \wedge \text{WF}_{\langle \text{target} \rangle}(\text{StopTarget}) \\
\text{IsOrderedChange}(p, i) & \triangleq \\
& \wedge \text{history}[i].\text{type} = \text{Change} \\
& \wedge \text{history}[i].\text{phase} = p \\
& \wedge \neg \exists j \in \text{DOMAIN } \text{history} : \\
& \quad \wedge j < i \\
& \quad \wedge \text{history}[j].\text{type} = \text{Change} \\
& \quad \wedge \text{history}[j].\text{phase} = p \\
& \quad \wedge \text{history}[j].\text{index} \geq \text{history}[i].\text{index} \\
\text{IsOrderedRollback}(p, i) & \triangleq \\
& \wedge \text{history}[i].\text{type} = \text{Rollback} \\
& \wedge \text{history}[i].\text{phase} = p \\
& \wedge \neg \exists j \in \text{DOMAIN } \text{history} : \\
& \quad \wedge j < i \\
& \quad \wedge \text{history}[j].\text{type} = \text{Change} \\
& \quad \wedge \text{history}[j].\text{phase} = p \\
& \quad \wedge \text{history}[j].\text{index} > \text{history}[i].\text{index} \\
& \quad \wedge \neg \exists k \in \text{DOMAIN } \text{history} : \\
& \quad \quad \wedge k > j \\
& \quad \quad \wedge k < i \\
& \quad \quad \wedge \text{history}[k].\text{type} = \text{Rollback} \\
& \quad \quad \wedge \text{history}[k].\text{phase} = p \\
& \quad \quad \wedge \text{history}[k].\text{index} = \text{history}[j].\text{index} \\
\text{Order} & \triangleq \\
& \wedge \forall i \in \text{DOMAIN } \text{history} : \\
& \quad \vee \text{IsOrderedChange}(\text{Commit}, i) \\
& \quad \vee \text{IsOrderedChange}(\text{Apply}, i) \\
& \quad \vee \text{IsOrderedRollback}(\text{Commit}, i) \\
& \quad \vee \text{IsOrderedRollback}(\text{Apply}, i) \\
& \wedge \forall i \in \text{DOMAIN } \text{proposal} : \\
& \quad \wedge \text{proposal}[i].\text{change.apply} = \text{Failed} \\
& \quad \wedge \text{proposal}[i].\text{rollback.apply} \neq \text{Complete} \\
& \quad \Rightarrow \forall j \in \text{DOMAIN } \text{proposal} : j > i \Rightarrow \\
& \quad \quad \text{proposal}[j].\text{change.apply} \in \{\text{None}, \text{Pending}, \text{Aborted}\} \\
\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}
\end{aligned}$$

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

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

THEOREM $\text{Spec} \Rightarrow \text{Safety}$

$\text{Termination} \triangleq$

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

$$\text{Liveness} \triangleq \text{Termination}$$

THEOREM $\text{Spec} \Rightarrow \text{Liveness}$

\ * Modification History
\ * Last modified *Fri Apr 21 18:30:03 PDT 2023* by *jhalterm*
\ * Last modified *Mon Feb 21 01:32:07 PST 2022* by *jordanhalterm*
\ * Created *Wed Sep 22 13:22:32 PDT 2021* by *jordanhalterm*