
MODULE *Config*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

GenerateTestCases \triangleq FALSE

Nil \triangleq "<nil>"

Change \triangleq "Change"

Rollback \triangleq "Rollback"

Commit \triangleq "Commit"

Apply \triangleq "Apply"

Pending \triangleq "Pending"

InProgress \triangleq "InProgress"

Complete \triangleq "Complete"

Aborted \triangleq "Aborted"

Failed \triangleq "Failed"

Done \triangleq { *Complete*, *Aborted*, *Failed* }

Node \triangleq { "node1" }

NumTransactions \triangleq 3

Path \triangleq { "path1" }

Value \triangleq { "value1", "value2" }

A transaction log. Transactions may either request a set of changes to a set of targets or rollback a prior change.

VARIABLE *transaction*

A record of per-target proposals

VARIABLE *proposal*

A record of per-target configurations

VARIABLE *configuration*

A record of target states

VARIABLE *target*

A record of target masterships

VARIABLE *mastership*

A sequence of state changes used for model checking.

VARIABLE *history*

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

LOCAL *Transaction* \triangleq INSTANCE *Transaction*

LOCAL *Proposal* \triangleq INSTANCE *Proposal*

LOCAL *Configuration* \triangleq INSTANCE *Configuration*

LOCAL *Mastership* \triangleq INSTANCE *Mastership*

$\text{RequestChange}(p, v) \triangleq$
 $\wedge \text{Transaction!RequestChange}(p, v)$

$\text{RequestRollback}(i) \triangleq$
 $\wedge \text{Transaction!RequestRollback}(i)$

$\text{SetMaster}(n) \triangleq$
 $\wedge \text{Mastership!SetMaster}(n)$
 $\wedge \text{UNCHANGED} \langle \text{transaction}, \text{proposal}, \text{configuration}, \text{target}, \text{history} \rangle$

$\text{UnsetMaster} \triangleq$
 $\wedge \text{Mastership!UnsetMaster}$
 $\wedge \text{UNCHANGED} \langle \text{transaction}, \text{proposal}, \text{configuration}, \text{target}, \text{history} \rangle$

$\text{ReconcileTransaction}(n, i) \triangleq$
 $\wedge i \in \text{DOMAIN } \text{transaction}$
 $\wedge \text{Transaction!ReconcileTransaction}(n, i)$
 $\wedge \text{GenerateTestCases} \Rightarrow \text{Transaction!Test!Log}([node \mapsto n, index \mapsto i])$

$\text{ReconcileProposal}(n, i) \triangleq$
 $\wedge i \in \text{DOMAIN } \text{proposal}$
 $\wedge \text{Proposal!ReconcileProposal}(n, i)$
 $\wedge \text{UNCHANGED} \langle \text{transaction} \rangle$
 $\wedge \text{GenerateTestCases} \Rightarrow \text{Proposal!Test!Log}([node \mapsto n, index \mapsto i])$

$\text{ReconcileConfiguration}(n) \triangleq$
 $\wedge \text{Configuration!ReconcileConfiguration}(n)$
 $\wedge \text{UNCHANGED} \langle \text{transaction}, \text{proposal}, \text{history} \rangle$
 $\wedge \text{GenerateTestCases} \Rightarrow \text{Configuration!Test!Log}([node \mapsto n])$

Formal specification, constraints, and theorems.

$$\begin{aligned}
Init &\triangleq \\
&\wedge transaction = [\\
&\quad i \in \{\} \mapsto [\\
&\quad \quad type \mapsto Change, \\
&\quad \quad index \mapsto 0, \\
&\quad \quad values \mapsto [p \in \{\} \mapsto Nil], \\
&\quad \quad commit \mapsto Pending, \\
&\quad \quad apply \mapsto Pending]] \\
&\wedge proposal = [\\
&\quad i \in \{\} \mapsto [\\
&\quad \quad change \mapsto [\\
&\quad \quad \quad phase \mapsto Nil, \\
&\quad \quad \quad state \mapsto Nil, \\
&\quad \quad \quad values \mapsto [\\
&\quad \quad \quad \quad p \in \{\} \mapsto [\\
&\quad \quad \quad \quad \quad index \mapsto 0, \\
&\quad \quad \quad \quad \quad value \mapsto Nil]]], \\
&\quad \quad rollback \mapsto [\\
&\quad \quad \quad phase \mapsto Nil, \\
&\quad \quad \quad state \mapsto Nil, \\
&\quad \quad \quad values \mapsto [\\
&\quad \quad \quad \quad p \in \{\} \mapsto [\\
&\quad \quad \quad \quad \quad index \mapsto 0, \\
&\quad \quad \quad \quad \quad value \mapsto Nil]]]]] \\
&\wedge configuration = [\\
&\quad state \mapsto InProgress, \\
&\quad term \mapsto 0, \\
&\quad committed \mapsto [\\
&\quad \quad index \mapsto 0, \\
&\quad \quad revision \mapsto 0, \\
&\quad \quad values \mapsto [\\
&\quad \quad \quad p \in \{\} \mapsto [\\
&\quad \quad \quad \quad index \mapsto 0, \\
&\quad \quad \quad \quad value \mapsto Nil]]], \\
&\quad applied \mapsto [\\
&\quad \quad index \mapsto 0, \\
&\quad \quad revision \mapsto 0, \\
&\quad \quad values \mapsto [\\
&\quad \quad \quad p \in \{\} \mapsto [\\
&\quad \quad \quad \quad index \mapsto 0, \\
&\quad \quad \quad \quad value \mapsto Nil]]]]] \\
&\wedge target = [
\end{aligned}$$

$$\begin{aligned}
& \text{values} \mapsto [\\
& \quad p \in \{\} \mapsto [\\
& \quad \quad \text{index} \mapsto 0, \\
& \quad \quad \text{value} \mapsto \text{Nil}]]] \\
& \wedge \text{mastership} = [\\
& \quad \text{master} \mapsto \text{Nil}, \\
& \quad \text{term} \mapsto 0] \\
& \wedge \text{history} = \langle \rangle \\
\text{Next} & \triangleq \\
& \vee \exists p \in \text{Path}, v \in \text{Value} : \\
& \quad \text{RequestChange}(p, v) \\
& \vee \exists i \in \text{DOMAIN transaction} : \\
& \quad \text{RequestRollback}(i) \\
& \vee \exists n \in \text{Node} : \\
& \quad \text{SetMaster}(n) \\
& \vee \exists t \in \text{DOMAIN Target} : \\
& \quad \text{UnsetMaster}(t) \\
& \vee \exists n \in \text{Node} : \\
& \quad \exists i \in \text{DOMAIN transaction} : \\
& \quad \quad \text{ReconcileTransaction}(n, i) \\
& \vee \exists n \in \text{Node} : \\
& \quad \exists i \in \text{DOMAIN proposal} : \\
& \quad \quad \text{ReconcileProposal}(n, i) \\
& \vee \exists n \in \text{Node} : \\
& \quad \text{ReconcileConfiguration}(n) \\
\text{Spec} & \triangleq \\
& \wedge \text{Init} \\
& \wedge \Box[\text{Next}]_{\text{vars}} \\
& \wedge \forall p \in \text{Path}, v \in \text{Value} : \\
& \quad \text{WF}_{\langle \text{transaction}, \text{proposal}, \text{configuration}, \text{mastership}, \text{target} \rangle}(\text{Transaction!RequestChange}(p, v)) \\
& \wedge \forall i \in 1 \dots \text{NumTransactions} : i \in \text{DOMAIN transaction} \Rightarrow \\
& \quad \text{WF}_{\langle \text{transaction}, \text{proposal}, \text{configuration}, \text{mastership}, \text{target} \rangle}(\text{Transaction!RequestRollback}(i)) \\
& \wedge \forall n \in \text{Node} : \\
& \quad \text{WF}_{\langle \text{mastership} \rangle}(\text{Mastership!SetMaster}(n)) \\
& \wedge \exists t \in \text{DOMAIN Target} : \\
& \quad \text{WF}_{\langle \text{mastership} \rangle}(\text{Mastership!UnsetMaster}(t)) \\
& \wedge \forall n \in \text{Node}, i \in 1 \dots \text{NumTransactions} : \\
& \quad \text{WF}_{\langle \text{transaction}, \text{proposal}, \text{configuration}, \text{mastership}, \text{target} \rangle}(\text{Transaction!ReconcileTransaction}(n, i)) \\
& \wedge \forall n \in \text{Node}, i \in 1 \dots \text{NumTransactions} : \\
& \quad \text{WF}_{\langle \text{proposal}, \text{configuration}, \text{mastership}, \text{target}, \text{history} \rangle}(\text{Proposal!ReconcileProposal}(n, i)) \\
& \wedge \forall n \in \text{Node} : \\
& \quad \text{WF}_{\langle \text{configuration}, \text{mastership}, \text{target} \rangle}(\text{Configuration!ReconcileConfiguration}(n))
\end{aligned}$$

$$LimitTransactions \triangleq Len(transaction) \leq NumTransactions$$

$$TypeOK \triangleq$$

$$\begin{aligned} & \wedge Transaction! TypeOK \\ & \wedge Proposal! TypeOK \\ & \wedge Configuration! TypeOK \\ & \wedge Mastership! TypeOK \end{aligned}$$

$$LOCAL \ IsOrderedChange(p, i) \triangleq$$

$$\begin{aligned} & \wedge history[i].type = Change \\ & \wedge history[i].phase = p \\ & \wedge \neg \exists j \in DOMAIN \ history : \\ & \quad \wedge j < i \\ & \quad \wedge history[j].type = Change \\ & \quad \wedge history[j].phase = p \\ & \quad \wedge history[j].index \geq history[i].index \end{aligned}$$

$$LOCAL \ IsOrderedRollback(p, i) \triangleq$$

$$\begin{aligned} & \wedge history[i].type = Rollback \\ & \wedge history[i].phase = p \\ & \wedge \neg \exists j \in DOMAIN \ history : \\ & \quad \wedge j < i \\ & \quad \wedge history[j].type = Change \\ & \quad \wedge history[j].phase = p \\ & \quad \wedge history[j].index > history[i].index \\ & \wedge \neg \exists k \in DOMAIN \ history : \\ & \quad \wedge k > j \\ & \quad \wedge k < i \\ & \quad \wedge history[k].type = Rollback \\ & \quad \wedge history[k].phase = p \\ & \quad \wedge history[k].index = history[j].index \end{aligned}$$

$$Order \triangleq$$

$$\begin{aligned} & \wedge \forall i \in DOMAIN \ history : \\ & \quad \vee IsOrderedChange(Commit, i) \\ & \quad \vee IsOrderedChange(Apply, i) \\ & \quad \vee IsOrderedRollback(Commit, i) \\ & \quad \vee IsOrderedRollback(Apply, i) \\ & \wedge \forall i \in DOMAIN \ proposal : \\ & \quad \wedge proposal[i].change.phase = Apply \\ & \quad \wedge proposal[i].change.state = Failed \\ & \quad \wedge proposal[i].rollback.phase = Apply \Rightarrow proposal[i].rollback.state \neq Complete \\ & \Rightarrow \forall j \in DOMAIN \ proposal : (j > i \Rightarrow \\ & \quad (proposal[j].change.phase = Apply \Rightarrow \end{aligned}$$

$$proposal[j].change.state \in \{Nil, Pending, Aborted\}))$$

Consistency \triangleq

$$\begin{aligned}
& \wedge \forall i \in \text{DOMAIN } proposal : \\
& \quad \vee configuration.committed.index < i \\
& \quad \vee configuration.committed.revision < i \\
& \quad \Rightarrow \neg \exists p \in \text{DOMAIN } configuration.committed.values : \\
& \quad \quad configuration.committed.values[p].index = i \\
& \wedge \forall i \in \text{DOMAIN } proposal : \\
& \quad \vee configuration.applied.index < i \\
& \quad \vee configuration.applied.revision < i \\
& \quad \Rightarrow \wedge \neg \exists p \in \text{DOMAIN } configuration.applied.values : \\
& \quad \quad configuration.applied.values[p].index = i \\
& \quad \wedge \neg \exists p \in \text{DOMAIN } target.values : \\
& \quad \quad target.values[p].index = i \\
& \wedge configuration.state = Complete \Rightarrow \\
& \quad \forall i \in \text{DOMAIN } proposal : \\
& \quad \quad \wedge configuration.applied.index \geq i \\
& \quad \quad \wedge configuration.applied.revision \geq i \\
& \quad \quad \Rightarrow \forall p \in \text{DOMAIN } proposal[i].change.values : \\
& \quad \quad \quad \wedge \neg \exists j \in \text{DOMAIN } proposal : \\
& \quad \quad \quad \quad \wedge j > i \\
& \quad \quad \quad \quad \wedge configuration.applied.index \geq j \\
& \quad \quad \quad \quad \wedge configuration.applied.revision \geq j \\
& \quad \quad \Rightarrow \wedge p \in \text{DOMAIN } target.values \\
& \quad \quad \quad \wedge target.values[p].value = proposal[i].change.values[p].value \\
& \quad \quad \quad \wedge target.values[p].index = proposal[i].change.values[p].index
\end{aligned}$$

Safety $\triangleq \Box (Order \wedge Consistency)$

THEOREM $Spec \Rightarrow Safety$

Terminates(i) \triangleq

$$\begin{aligned}
& \wedge i \in \text{DOMAIN } transaction \\
& \wedge transaction[i].commit \in Done \\
& \wedge transaction[i].apply \in Done
\end{aligned}$$

Termination \triangleq

$$\forall i \in 1 \dots NumTransactions : \Diamond Terminates(i)$$

Liveness $\triangleq Termination$

THEOREM $Spec \Rightarrow Liveness$