——————————————— 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 $target$

1

A record of target masterships
VARIABLE *mastership*

A sequence of state changes used for model checking.
VARIABLE *history*

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

LOCAL $Transaction \triangleq$ INSTANCE $Transaction$

LOCAL $Proposal \triangleq$ INSTANCE $Proposal$

LOCAL $Configuration \triangleq$ INSTANCE $Configuration$

LOCAL $Mastership \triangleq$ INSTANCE $Mastership$

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

$RequestRollback(i) \triangleq$
    $\wedge \, Transaction\,!\,RequestRollback(i)$

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

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

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

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

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

2

Formal specification, constraints, and theorems.

$Init \triangleq$

$\quad \wedge \; transaction = [$

$\qquad i \in \{\} \mapsto [$

$\qquad\quad type \quad\;\; \mapsto Change,$

$\qquad\quad index \quad \mapsto 0,$

$\qquad\quad values \quad \mapsto [p \in \{\} \mapsto Nil],$

$\qquad\quad commit \mapsto Pending,$

$\qquad\quad apply \quad\; \mapsto Pending]]$

$\quad \wedge \; proposal = [$

$\qquad i \in \{\} \mapsto [$

$\qquad\quad change \mapsto [$

$\qquad\qquad phase \;\; \mapsto Nil,$

$\qquad\qquad state \quad \mapsto Nil,$

$\qquad\qquad values \; \mapsto [$

$\qquad\qquad\quad p \in \{\} \;\; \mapsto [$

$\qquad\qquad\qquad index \mapsto 0,$

$\qquad\qquad\qquad value \mapsto Nil]]],$

$\qquad\quad rollback \mapsto [$

$\qquad\qquad phase \;\; \mapsto Nil,$

$\qquad\qquad state \quad \mapsto Nil,$

$\qquad\qquad values \; \mapsto [$

$\qquad\qquad\quad p \quad \in \{\} \;\; \mapsto [$

$\qquad\qquad\qquad index \mapsto 0,$

$\qquad\qquad\qquad value \mapsto Nil]]]]]$

$\quad \wedge \; configuration = [$

$\qquad state \;\; \mapsto InProgress,$

$\qquad term \;\; \mapsto 0,$

$\qquad committed \mapsto [$

$\qquad\quad index \qquad \mapsto 0,$

$\qquad\quad revision \; \mapsto 0,$

$\qquad\quad values \quad\;\; \mapsto [$

$\qquad\qquad p \in \{\} \;\; \mapsto [$

$\qquad\qquad\quad index \mapsto 0,$

$\qquad\qquad\quad value \mapsto Nil]]],$

$\qquad applied \mapsto [$

$\qquad\quad index \qquad \mapsto 0,$

$\qquad\quad revision \; \mapsto 0,$

$\qquad\quad values \quad\;\; \mapsto [$

$\qquad\qquad p \in \{\} \;\; \mapsto [$

$\qquad\qquad\quad index \mapsto 0,$

$\qquad\qquad\quad value \mapsto Nil]]]]$

$\quad \wedge \; target = [$

3

$$
\begin{aligned}
&\quad values \mapsto [\\
&\qquad p \in \{\} \;\mapsto [\\
&\qquad\quad index \mapsto 0, \\
&\qquad\quad value \mapsto Nil]]] \\
&\land mastership = [\\
&\quad master \mapsto Nil, \\
&\quad term \quad \mapsto 0] \\
&\land history = \langle\rangle
\end{aligned}
$$

$Next \;\triangleq$
$\quad \lor \exists\, p \in Path,\, v \in Value :$
$\qquad RequestChange(p,\, v)$
$\quad \lor \exists\, i \in \text{DOMAIN } transaction :$
$\qquad RequestRollback(i)$
$\quad \lor \exists\, n \in Node :$
$\qquad SetMaster(n)$
$\quad\; \lor \exists\, t \in \text{DOMAIN } \; Target :$
$\qquad UnsetMaster(t)$
$\quad \lor \exists\, n \in Node :$
$\qquad \exists\, i \in \text{DOMAIN } transaction :$
$\qquad\quad ReconcileTransaction(n,\, i)$
$\quad \lor \exists\, n \in Node :$
$\qquad \exists\, i \in \text{DOMAIN } proposal :$
$\qquad\quad ReconcileProposal(n,\, i)$
$\quad \lor \exists\, n \in Node :$
$\qquad ReconcileConfiguration(n)$

$Spec \;\triangleq$
$\quad \land Init$
$\quad \land \Box[Next]_{vars}$
$\quad \land \forall\, p \in Path,\, v \in Value :$
$\qquad \text{WF}_{\langle transaction,\, proposal,\, configuration,\, mastership,\, target \rangle}(Transaction\,!\,RequestChange(p,\, v))$
$\quad \land \forall\, i \in 1\,..\,NumTransactions : i \in \text{DOMAIN } transaction \Rightarrow$
$\qquad \text{WF}_{\langle transaction,\, proposal,\, configuration,\, mastership,\, target \rangle}(Transaction\,!\,RequestRollback(i))$
$\quad \land \forall\, n \in Node :$
$\qquad \text{WF}_{\langle mastership \rangle}(Mastership\,!\,SetMaster(n))$
$\quad\; \land \exists\, t \in \text{DOMAIN } \; Target :$
$\qquad \text{WF}_{\_\langle mastership \rangle}(Mastership\,!\,UnsetMaster(t))$
$\quad \land \forall\, n \in Node,\, i \in 1\,..\,NumTransactions :$
$\qquad \text{WF}_{\langle transaction,\, proposal,\, configuration,\, mastership,\, target \rangle}(Transaction\,!\,ReconcileTransaction(n,\, i))$
$\quad \land \forall\, n \in Node,\, i \in 1\,..\,NumTransactions :$
$\qquad \text{WF}_{\langle proposal,\, configuration,\, mastership,\, target,\, history \rangle}(Proposal\,!\,ReconcileProposal(n,\, i))$
$\quad \land \forall\, n \in Node :$
$\qquad \text{WF}_{\langle configuration,\, mastership,\, target \rangle}(Configuration\,!\,ReconcileConfiguration(n))$

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

$TypeOK \triangleq$
 $\land Transaction!TypeOK$
 $\land Proposal!TypeOK$
 $\land Configuration!TypeOK$
 $\land Mastership!TypeOK$

LOCAL $IsOrderedChange(p,\,i) \triangleq$
 $\land\quad history[i].type = Change$
 $\land\quad history[i].phase = p$
 $\land\quad \neg\exists\, j \in \text{DOMAIN } history :$
   $\land\, j < i$
   $\land\, history[j].type = Change$
   $\land\, history[j].phase = p$
   $\land\, history[j].index \geq history[i].index$

LOCAL $IsOrderedRollback(p,\,i) \triangleq$
 $\land\quad history[i].type = Rollback$
 $\land\quad history[i].phase = p$
 $\land\quad \neg\exists\, j \in \text{DOMAIN } history :$
   $\land\, j < i$
   $\land\, history[j].type = Change$
   $\land\, history[j].phase = p$
   $\land\, history[j].index > history[i].index$
   $\land\, \neg\exists\, k \in \text{DOMAIN } history :$
    $\land\, k > j$
    $\land\, k < i$
    $\land\, history[k].type = Rollback$
    $\land\, history[k].phase = p$
    $\land\, history[k].index = history[j].index$

$Order \triangleq$
 $\land\ \forall\, i \in \text{DOMAIN } history :$
  $\lor IsOrderedChange(Commit,\,i)$
  $\lor IsOrderedChange(Apply,\,i)$
  $\lor IsOrderedRollback(Commit,\,i)$
  $\lor IsOrderedRollback(Apply,\,i)$
 $\land\ \forall\, i \in \text{DOMAIN } proposal :$
  $\land\, proposal[i].change.phase = Apply$
  $\land\, proposal[i].change.state\ = Failed$
  $\land\, proposal[i].rollback.phase = Apply \Rightarrow proposal[i].rollback.state \neq Complete$
  $\Rightarrow \forall\, j \in \text{DOMAIN } proposal : (j > i \Rightarrow$
   $(proposal[j].change.phase = Apply \Rightarrow$

5

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

$Consistency \triangleq$
 $\land\ \forall\, i \in \text{DOMAIN}\ proposal :$
  $\lor\ configuration.committed.index < i$
  $\lor\ configuration.committed.revision < i$
  $\Rightarrow \neg\exists\, p \in \text{DOMAIN}\ configuration.committed.values :$
   $configuration.committed.values[p].index = i$
 $\land\ \forall\, i \in \text{DOMAIN}\ proposal :$
  $\lor\ configuration.applied.index < i$
  $\lor\ configuration.applied.revision < i$
  $\Rightarrow\ \land\ \neg\exists\, p \in \text{DOMAIN}\ configuration.applied.values :$
    $configuration.applied.values[p].index = i$
   $\land\ \neg\exists\, p \in \text{DOMAIN}\ target.values :$
    $target.values[p].index = i$
 $\land\ configuration.state = Complete \Rightarrow$
  $\forall\, i \in \text{DOMAIN}\ proposal :$
   $\land\ configuration.applied.index \geq i$
   $\land\ configuration.applied.revision \geq i$
   $\Rightarrow \forall\, p \in \text{DOMAIN}\ proposal[i].change.values :$
    $\land\ \neg\exists\, j \in \text{DOMAIN}\ proposal :$
     $\land\ j > i$
     $\land\ configuration.applied.index \geq j$
     $\land\ configuration.applied.revision \geq j$
    $\Rightarrow\ \land\ p \in \text{DOMAIN}\ target.values$
     $\land\ target.values[p].value = proposal[i].change.values[p].value$
     $\land\ target.values[p].index = proposal[i].change.values[p].index$

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

THEOREM $Spec \Rightarrow Safety$

$Termination \triangleq$
 $\forall\, i \in 1\,..\,NumTransactions :$
  $\land\ proposal[transaction[i].index].change.phase = Commit \rightsquigarrow$
   $\land\ proposal[transaction[i].index].change.phase = Commit$
   $\land\ proposal[transaction[i].index].change.state\ \in Done$
  $\land\ proposal[transaction[i].index].change.phase = Apply \rightsquigarrow$
   $\land\ proposal[transaction[i].index].change.phase = Apply$
   $\land\ proposal[transaction[i].index].change.apply\ \in Done$
  $\land\ proposal[transaction[i].index].rollback.phase = Commit \rightsquigarrow$
   $\land\ proposal[transaction[i].index].rollback.phase = Commit$
   $\land\ proposal[transaction[i].index].rollback.commit\ \in Done$
  $\land\ proposal[transaction[i].index].rollback.phase = Apply \rightsquigarrow$
   $\land\ proposal[transaction[i].index].rollback.phase = Apply$
   $\land\ proposal[transaction[i].index].rollback.apply\ \in Done$

$Liveness \triangleq Termination$

THEOREM $Spec \Rightarrow Liveness$