```
──────────────────── MODULE Config ────────────────────
INSTANCE Naturals

INSTANCE FiniteSets

INSTANCE Sequences

INSTANCE TLC
───────────────────────────────────────────────────────
GenerateTestCases ≜ FALSE

Nil ≜ "<nil>"

Change ≜ "Change"
Rollback ≜ "Rollback"

Commit ≜ "Commit"
Apply ≜ "Apply"

Pending ≜ "Pending"
Complete ≜ "Complete"
Aborted ≜ "Aborted"
Failed ≜ "Failed"

Node ≜ {"node1"}

NumTransactions ≜ 3
NumTerms ≜ 2
NumConns ≜ 2
NumStarts ≜ 2

Path ≜ {"path1"}
Value ≜ {"value1", "value2"}
───────────────────────────────────────────────────────
  A transaction log of changes and rollbacks.
VARIABLE transaction

  A record of per-target configurations
VARIABLE configuration

  A record of target masterships
VARIABLE mastership

  A record of node connections to the target
VARIABLE conn
```

The target state

VARIABLE target

A sequence of state changes used for model checking.

VARIABLE $\mathit{history}$

$\mathit{vars} \triangleq \langle \mathit{transaction}, \mathit{configuration}, \mathit{mastership}, \mathit{conn}, \mathit{target}, \mathit{history} \rangle$

LOCAL $\mathit{Transaction} \triangleq$ INSTANCE $\mathit{Transaction}$

LOCAL $\mathit{Configuration} \triangleq$ INSTANCE $\mathit{Configuration}$

LOCAL $\mathit{Mastership} \triangleq$ INSTANCE $\mathit{Mastership}$

LOCAL $\mathit{Target} \triangleq$ INSTANCE Target

$\mathit{AppendChange}(i) \triangleq$
$\quad \land \mathit{Transaction}!\mathit{AppendChange}(i)$

$\mathit{RollbackChange}(i) \triangleq$
$\quad \land \mathit{Transaction}!\mathit{RollbackChange}(i)$

$\mathit{ReconcileTransaction}(n, i) \triangleq$
$\quad \land \mathit{Transaction}!\mathit{ReconcileTransaction}(n, i)$
$\quad \land \mathit{GenerateTestCases} \Rightarrow \mathit{Transaction}!\mathit{Test}!\mathit{Log}([\mathit{node} \mapsto n, \mathit{index} \mapsto i])$

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

$\mathit{ReconcileMastership}(n) \triangleq$
$\quad \land \mathit{Mastership}!\mathit{ReconcileMastership}(n)$
$\quad \land$ UNCHANGED $\langle \mathit{transaction}, \mathit{configuration}, \mathit{target}, \mathit{history} \rangle$
$\quad \land \mathit{GenerateTestCases} \Rightarrow \mathit{Mastership}!\mathit{Test}!\mathit{Log}([\mathit{node} \mapsto n])$

$\mathit{ConnectNode}(n) \triangleq$
$\quad \land \mathit{Target}!\mathit{Connect}(n)$
$\quad \land$ UNCHANGED $\langle \mathit{transaction}, \mathit{configuration}, \mathit{mastership}, \mathit{history} \rangle$

$\mathit{DisconnectNode}(n) \triangleq$
$\quad \land \mathit{Target}!\mathit{Disconnect}(n)$
$\quad \land$ UNCHANGED $\langle \mathit{transaction}, \mathit{configuration}, \mathit{mastership}, \mathit{history} \rangle$

$\mathit{StartTarget} \triangleq$
$\quad \land \mathit{Target}!\mathit{Start}$

```
    $\land$ UNCHANGED $\langle transaction, configuration, mastership, history \rangle$

$StopTarget \triangleq$
    $\land Target!Stop$
    $\land$ UNCHANGED $\langle transaction, configuration, mastership, history \rangle$
```

Formal specification, constraints, and theorems.

```
$Init \triangleq$
    $\land transaction = [$
          $i \in \{\} \mapsto [$
            $type \mapsto Nil,$
            $index \mapsto 0,$
            $revision \mapsto 0,$
            $commit \mapsto Nil,$
            $apply \mapsto Nil,$
            $change \mapsto [$
                $index \mapsto 0,$
                $revision \mapsto 0,$
                $values \mapsto [$
                    $p \in \{\} \mapsto [$
                        $index \mapsto 0,$
                        $value \mapsto Nil]]],$
            $rollback \mapsto [$
                $index \mapsto 0,$
                $revision \mapsto 0,$
                $values \mapsto [$
                    $p \in \{\} \mapsto [$
                        $index \mapsto 0,$
                        $value \mapsto Nil]]]]]$
    $\land configuration = [$
          $state \mapsto Pending,$
          $term \mapsto 0,$
          $committed \mapsto [$
              $index \mapsto 0,$
              $revision \mapsto 0,$
              $values \mapsto [$
                  $p \in \{\} \mapsto [$
                      $index \mapsto 0,$
                      $value \mapsto Nil]]],$
          $applied \mapsto [$
              $target \mapsto 0,$
              $index \mapsto 0,$
              $revision \mapsto 0,$
              $values \mapsto [$
```

```
                $p \in \{\} \ \mapsto [$
                    $index \mapsto 0,$
                    $value \ \mapsto Nil]]]]$
    $\land target = [$
            $id \qquad\quad \mapsto 1,$
            $running \mapsto \text{TRUE},$
            $values \quad \mapsto [$
                $p \in \{\} \ \mapsto [$
                    $index \mapsto 0,$
                    $value \ \mapsto Nil]]]$
    $\land mastership = [$
            $master \mapsto \text{CHOOSE } n \in Node : \text{TRUE},$
            $term \quad \mapsto 1,$
            $conn \quad \mapsto 1]$
    $\land conn = [$
            $n \ \in Node \mapsto [$
                $id \qquad\quad \mapsto 1,$
                $connected \mapsto \text{TRUE}]]$
    $\land history = \langle\rangle$

$Next \ \triangleq$
    $\lor \exists\, i \in 1 \,..\, NumTransactions :$
          $\lor AppendChange(i)$
          $\lor RollbackChange(i)$
    $\lor \exists\, n \in Node :$
          $\exists\, i \in \text{DOMAIN } transaction :$
            $ReconcileTransaction(n, i)$
    $\lor \exists\, n \in Node :$
          $ReconcileConfiguration(n)$
    $\lor \exists\, n \in Node :$
          $ReconcileMastership(n)$
    $\lor \exists\, n \in Node :$
          $\lor ConnectNode(n)$
          $\lor DisconnectNode(n)$
    $\lor StartTarget$
    $\lor StopTarget$

$Spec \ \triangleq$
    $\land Init$
    $\land \Box[Next]_{vars}$
    $\land \forall\, i \in 1 \,..\, NumTransactions :$
          $\text{WF}_{\langle transaction \rangle}(Transaction!RollbackChange(i))$
    $\land \forall\, n \in Node :$
          $\text{WF}_{vars}(\exists\, i \in \text{DOMAIN } transaction : Transaction!ReconcileTransaction(n, i))$
    $\land \forall\, n \in Node :$
```

```
        $\mathrm{WF}_{\langle configuration,\, mastership,\, conn,\, target \rangle}(Configuration!ReconcileConfiguration(n))$
    $\land \forall\, n \in Node :$
        $\mathrm{WF}_{\langle mastership,\, conn \rangle}(Mastership!ReconcileMastership(n))$
    $\land \forall\, n \in Node :$
        $\mathrm{WF}_{\langle conn,\, target \rangle}(Target!Connect(n) \lor Target!Disconnect(n))$
    $\land \mathrm{WF}_{\langle conn,\, target \rangle}(Target!Start \lor Target!Stop)$
```

```
$LimitTerms \triangleq$
    $\lor mastership.term < NumTerms$
    $\lor \land mastership.term = NumTerms$
      $\land mastership.master \neq Nil$

$LimitConns \triangleq$
    $\forall\, n \in \textsc{domain}\ conn :$
      $\lor conn[n].id < NumConns$
      $\lor \land conn[n].id = NumConns$
        $\land conn[n].connected$

$LimitStarts \triangleq$
    $\lor target.id < 2$
    $\lor \land target.id = 2$
      $\land target.running$
```

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

$\textsc{local}\ IsOrderedChange(p,\, i) \triangleq$
    $\land$   $history[i].type = Change$
    $\land$   $history[i].phase = p$
    $\land$   $\neg\exists\, j \in \textsc{domain}\ history :$
                $\land j < i$
                $\land history[j].type = Change$
                $\land history[j].phase = p$
                $\land history[j].revision \geq history[i].revision$

$\textsc{local}\ IsOrderedRollback(p,\, i) \triangleq$
    $\land$   $history[i].type = Rollback$
    $\land$   $history[i].phase = p$
    $\land$   $\exists\, j \in \textsc{domain}\ history :$
              $\land j < i$
              $\land history[j].type = Change$
```

```
          $\land\ history[j].revision = history[i].revision$
  $\land\quad \neg\exists\, j \in \textsc{domain}\ history :$
            $\land\ j < i$
            $\land\ history[j].type = Change$
            $\land\ history[j].phase = p$
            $\land\ history[j].revision > history[i].revision$
            $\land\ \neg\exists\, k \in \textsc{domain}\ history :$
                $\land\ k > j$
                $\land\ k < i$
                $\land\ history[k].type = Rollback$
                $\land\ history[k].phase = p$
                $\land\ history[k].revision = history[j].revision$

$Order\ \triangleq$
  $\land\ \forall\, i \in \textsc{domain}\ history :$
      $\lor\ IsOrderedChange(Commit,\ i)$
      $\lor\ IsOrderedChange(Apply,\ i)$
      $\lor\ IsOrderedRollback(Commit,\ i)$
      $\lor\ IsOrderedRollback(Apply,\ i)$
  $\land\ \forall\, i \in \textsc{domain}\ transaction :$
        $\land\ transaction[i].type = Change$
        $\land\ transaction[i].apply = Failed$
        $\land\ \neg\exists\, j \in \textsc{domain}\ transaction :$
            $\land\ transaction[j].type = Rollback$
            $\land\ transaction[j].rollback.revision = transaction[i].change.revision$
            $\land\ transaction[j].apply = Complete$
        $\Rightarrow \forall\, j \in \textsc{domain}\ transaction : (j > i \Rightarrow$
            $(transaction[j].type = Change \Rightarrow transaction[j].apply \neq Complete))$

$Consistency\ \triangleq$
  $\land\ \forall\, i \in \textsc{domain}\ transaction :$
      $\land\ transaction[i].commit = Complete$
      $\land\ \neg\exists\, j \in \textsc{domain}\ transaction :$
          $\land\ j > i$
          $\land\ transaction[j].commit = Complete$
      $\Rightarrow \forall\, p \in \textsc{domain}\ transaction[i].change.values :$
          $\land\ configuration.committed.values[p] = transaction[i].change.values[p]$
  $\land\ \forall\, i \in \textsc{domain}\ transaction :$
      $\land\ transaction[i].apply = Complete$
      $\land\ \neg\exists\, j \in \textsc{domain}\ transaction :$
          $\land\ j > i$
          $\land\ transaction[j].apply = Complete$
      $\Rightarrow \forall\, p \in \textsc{domain}\ transaction[i].change.values :$
          $\land\ configuration.applied.values[p] = transaction[i].change.values[p]$
          $\land\ \land\ target.running$
```

```
                    $\land\ configuration.applied.target = target.id$
                    $\land\ configuration.state = Complete$
                    $\Rightarrow target.values[p] = transaction[i].change.values[p]$

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

THEOREM $Spec \Rightarrow Safety$

LOCAL $IsChanging(i) \triangleq$
  $\exists\, j \in$ DOMAIN $transaction$ :
     $\land\ transaction[j].type = Change$
     $\land\ transaction[j].change.revision = i$

LOCAL $IsChanged(i) \triangleq$
  $\exists\, j \in$ DOMAIN $transaction$ :
     $\land\ transaction[j].type = Change$
     $\land\ transaction[j].change.revision = i$
     $\land\ transaction[j].commit \neq Pending$
     $\land\ transaction[j].apply \neq Pending$

LOCAL $IsRollingBack(i) \triangleq$
  $\exists\, j \in$ DOMAIN $transaction$ :
     $\land\ transaction[j].type = Rollback$
     $\land\ transaction[j].rollback.revision = i$

LOCAL $IsRolledBack(i) \triangleq$
  $\exists\, j \in$ DOMAIN $transaction$ :
     $\land\ transaction[j].type = Rollback$
     $\land\ transaction[j].rollback.revision = i$
     $\land\ transaction[j].commit \neq Pending$
     $\land\ transaction[j].apply \neq Pending$

$Terminates(i) \triangleq$
   $\land\ IsChanging(i) \leadsto IsChanged(i)$
   $\land\ IsRollingBack(i) \leadsto IsRolledBack(i)$

$Termination \triangleq$
  $\forall\, i \in 1 \,..\, NumTransactions : Terminates(i)$

$Liveness \triangleq Termination$

THEOREM $Spec \Rightarrow Liveness$
```