```
MODULE Config

INSTANCE Naturals

INSTANCE FiniteSets

INSTANCE Sequences

INSTANCE TLC

GenerateTestCases ≜ FALSE

Nil ≜ "<nil>"

Change ≜ "Change"
Rollback ≜ "Rollback"

Initialize ≜ "Initialize"
Validate ≜ "Validate"
Abort ≜ "Abort"
Commit ≜ "Commit"
Apply ≜ "Apply"

InProgress ≜ "InProgress"
Complete ≜ "Complete"
Failed ≜ "Failed"

Pending ≜ "Pending"
Validated   ≜ "Validated"
Committed ≜ "Committed"
Applied ≜ "Applied"
Aborted ≜ "Aborted"

Valid ≜ TRUE
Invalid ≜ FALSE

Success ≜ "Success"
Failure ≜ "Failure"

Node ≜ {"node1"}

NumTransactions ≜ 3

Path ≜ {"path1"}
Value ≜ {"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

vars ≜ ⟨transaction, proposal, configuration, mastership, target⟩
```

```
LOCAL Transaction ≜ INSTANCE Transaction

LOCAL Proposal ≜ INSTANCE Proposal

LOCAL Configuration ≜ INSTANCE Configuration

LOCAL Mastership ≜ INSTANCE Mastership
```

```
RequestChange(p, v) ≜
    ∧ Transaction!RequestChange(p, v)

RequestRollback(i) ≜
    ∧ Transaction!RequestRollback(i)

SetMaster(n) ≜
    ∧ Mastership!SetMaster(n)
    ∧ UNCHANGED ⟨transaction, proposal, configuration, target⟩

UnsetMaster ≜
    ∧ Mastership!UnsetMaster
    ∧ UNCHANGED ⟨transaction, proposal, configuration, target⟩

ReconcileTransaction(n, i) ≜
    ∧ i ∈ DOMAIN transaction
    ∧ Transaction!ReconcileTransaction(n, i)
    ∧ GenerateTestCases ⇒ Transaction!Test!Log([node ↦ n, index ↦ i])

ReconcileProposal(n, i) ≜
    ∧ i ∈ DOMAIN proposal
    $\land Proposal!ReconcileProposal(n, i)$
    $\land \textsc{unchanged} \langle transaction \rangle$
    $\land GenerateTestCases \Rightarrow Proposal!Test!Log([node \mapsto n, index \mapsto i])$

$ReconcileConfiguration(n) \triangleq$
    $\land Configuration!ReconcileConfiguration(n)$
    $\land \textsc{unchanged} \langle transaction, proposal \rangle$
    $\land GenerateTestCases \Rightarrow Configuration!Test!Log([node \mapsto n])$
```

Formal specification, constraints, and theorems.

```
$Init \triangleq$
    $\land transaction = [$
          $i \in \{\} \mapsto [$
            $type \mapsto Change,$
            $phase \mapsto Initialize,$
            $state \mapsto InProgress]]$
    $\land proposal = [$
          $i \in \{\} \mapsto [$
            $phase \mapsto Initialize,$
            $state \mapsto InProgress]]$
    $\land configuration = [$
          $state \mapsto InProgress,$
          $config \mapsto [$
              $index \mapsto 0,$
              $term \mapsto 0,$
              $values \mapsto [$
                  $path \in \{\} \mapsto [$
                      $path \mapsto path,$
                      $value \mapsto Nil,$
                      $index \mapsto 0,$
                      $deleted \mapsto \textsc{false}]]],$
          $proposal \mapsto [index \mapsto 0],$
          $commit \mapsto [index \mapsto 0],$
          $target \mapsto [$
              $index \mapsto 0,$
              $term \mapsto 0,$
              $values \mapsto [$
                  $path \in \{\} \mapsto [$
                      $path \mapsto path,$
                      $value \mapsto Nil,$
                      $index \mapsto 0,$
                      $deleted \mapsto \textsc{false}]]]]$
    $\land target = [path \in \{\} \mapsto [value \mapsto Nil]]$
    $\land mastership = [master \mapsto Nil, term \mapsto 0]$
```

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

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

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

```
$Order \triangleq$
    $\forall i \in \text{DOMAIN } proposal :$
      $\land \land proposal[i].phase = Commit$
          $\land proposal[i].state \ = InProgress$
```

```
        $\Rightarrow \neg\exists j \in \text{DOMAIN } proposal :$
                $\land j > i$
                $\land proposal[j].phase = Commit$
                $\land proposal[j].state = Complete$
    $\land \land proposal[i].phase = Apply$
        $\land proposal[i].state = InProgress$
        $\Rightarrow \neg\exists j \in \text{DOMAIN } proposal :$
                $\land j > i$
                $\land proposal[j].phase = Apply$
                $\land proposal[j].state = Complete$

$Consistency \triangleq$
  LET
      Compute the transaction indexes that have been applied to the target
    $targetIndexes \triangleq \{i \in \text{DOMAIN } transaction :$
                          $\land i \in \text{DOMAIN } proposal$
                          $\land proposal[i].phase = Apply$
                          $\land proposal[i].state = Complete$
                          $\land \neg\exists j \in \text{DOMAIN } transaction :$
                                $\land j > i$
                                $\land transaction[j].type = Rollback$
                                $\land transaction[j].rollback = i$
                                $\land transaction[j].phase = Apply$
                                $\land transaction[j].state = Complete\}$
      Compute the set of paths in the target that have been updated by transactions
    $appliedPaths \triangleq \text{UNION } \{\text{DOMAIN } proposal[i].change.values : i \in targetIndexes\}$
      Compute the highest index applied to the target for each path
    $pathIndexes \triangleq [p \in appliedPaths \mapsto \text{CHOOSE } i \in targetIndexes :$
                        $\forall j \in targetIndexes :$
                          $\land i \geq j$
                          $\land p \in \text{DOMAIN } proposal[i].change.values]$
      Compute the expected target configuration based on the last indexes applied
      to the target for each path.
    $expectedConfig \triangleq [p \in \text{DOMAIN } pathIndexes \mapsto proposal[pathIndexes[p]].change.values[p]]$
  IN
    $target = expectedConfig$

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

THEOREM $Spec \Rightarrow Safety$

$Terminated(i) \triangleq$
  $\land i \in \text{DOMAIN } transaction$
  $\land \lor \land transaction[i].phase = Apply$
          $\land transaction[i].state = Complete$
     $\lor transaction[i].state = Failed$
```

$\mathit{Termination} \triangleq$
$\quad \forall\, i \in 1 \,..\, \mathit{NumTransactions} : \Diamond \mathit{Terminated}(i)$

$\mathit{Liveness} \triangleq \mathit{Termination}$

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