MODULE *Proposal*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

An empty constant
CONSTANT *Nil*

Proposal phase constants
CONSTANTS
Change,
Rollback

Proposal phase constants
CONSTANTS
Commit,
Apply

Status constants
CONSTANTS
Pending,
Complete,
Aborted,
Failed

$Status \triangleq \{Pending, Complete, Aborted, Failed\}$

The set of all nodes
CONSTANT *Node*

The set of possible paths and values
CONSTANT *Path*, *Value*

$Empty \triangleq [p \in \{\} \mapsto Nil]$

Variables defined by other modules.
VARIABLES
configuration,
mastership,
conn,
target

A proposal log. Proposals may either request a set
of changes to a set of targets or rollback a prior change.
VARIABLE *proposals*

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

$$
\begin{array}{l}
\mathit{TypeOK} \triangleq \\
\quad \forall\, i \in \text{DOMAIN } \mathit{proposals} : \\
\qquad \land\ \mathit{proposals}[i].\mathit{type} \in \{\mathit{Change},\ \mathit{Rollback}\} \\
\qquad \land\ \mathit{proposals}[i].\mathit{index} \in \mathit{Nat} \\
\qquad \land\ \mathit{proposals}[i].\mathit{revision} \in \mathit{Nat} \\
\qquad \land\ \mathit{proposals}[i].\mathit{change}.\mathit{index} \in \mathit{Nat} \\
\qquad \land\ \mathit{proposals}[i].\mathit{change}.\mathit{revision} \in \mathit{Nat} \\
\qquad \land\ \forall\, p \in \text{DOMAIN } \mathit{proposals}[i].\mathit{change}.\mathit{values} : \\
\qquad\quad \mathit{proposals}[i].\mathit{change}.\mathit{values}[p] \neq \mathit{Nil} \Rightarrow \\
\qquad\qquad \mathit{proposals}[i].\mathit{change}.\mathit{values}[p] \in \text{STRING} \\
\qquad \land\ \mathit{proposals}[i].\mathit{rollback}.\mathit{index} \in \mathit{Nat} \\
\qquad \land\ \mathit{proposals}[i].\mathit{rollback}.\mathit{revision} \in \mathit{Nat} \\
\qquad \land\ \forall\, p \in \text{DOMAIN } \mathit{proposals}[i].\mathit{rollback}.\mathit{values} : \\
\qquad\quad \mathit{proposals}[i].\mathit{rollback}.\mathit{values}[p] \neq \mathit{Nil} \Rightarrow \\
\qquad\qquad \mathit{proposals}[i].\mathit{rollback}.\mathit{values}[p] \in \text{STRING} \\
\qquad \land\ \mathit{proposals}[i].\mathit{commit} \in \mathit{Status} \\
\qquad \land\ \mathit{proposals}[i].\mathit{apply} \in \mathit{Status}
\end{array}
$$

$$
\begin{array}{l}
\text{LOCAL } \mathit{State} \triangleq [\\
\quad \mathit{proposals} \quad\ \mapsto \mathit{proposals}, \\
\quad \mathit{configuration} \mapsto \mathit{configuration}]
\end{array}
$$

$$
\begin{array}{l}
\text{LOCAL } \mathit{Transitions} \triangleq \\
\quad \text{LET} \\
\qquad \mathit{indexes} \triangleq \{i \in \text{DOMAIN } \mathit{proposals}' : \\
\qquad\qquad\qquad i \in \text{DOMAIN } \mathit{proposals} \Rightarrow \mathit{proposals}'[i] \neq \mathit{proposals}[i]\} \\
\quad \text{IN} \\
\qquad [\mathit{proposals} \mapsto [i \in \mathit{indexes} \mapsto \mathit{proposals}'[i]]]
\end{array}
$$

$$
\begin{array}{l}
\mathit{Test} \triangleq \text{INSTANCE } \mathit{Test} \text{ WITH} \\
\quad \mathit{File} \leftarrow \text{“Proposal.log”}
\end{array}
$$

CHANGE [*index* = 1, *revision* = 1, *change* = (*index* = 1, *revision* = 1), *rollback* = (*index* = 0, *revision* = 0)] ← − *Change* revision 1
CHANGE [*index* = 2, *revision* = 2, *change* = (*index* = 2, *revision* = 2), *rollback* = (*index* = 1, *revision* = 1)]
CHANGE [*index* = 3, *revision* = 3, *change* = (*index* = 3, *revision* = 3), *rollback* = (*index* = 2, *revision* = 2)]

ROLLBACK [$index = 4$, $revision = 3$, $change = (index = 2,\ revision = 2)$, $rollback = (index = 3,\ revision = 3)$] $\leftarrow -$ Roll back revision 3 at index 3, leading to revision 2
ROLLBACK [$index = 5$, $revision = 3$, $change = (index = 1,\ revision = 1)$, $rollback = (index = 2,\ revision = 2)$]
CHANGE [$index = 6$, $revision = 4$, $change = (index = 6,\ revision = 4)$, $rollback = (index = 1,\ revision = 1)$]
CHANGE [$index = 7$, $revision = 5$, $change = (index = 7,\ revision = 5)$, $rollback = (index = 6,\ revision = 4)$]
ROLLBACK [$index = 8$, $revision = 5$, $change = (index = 6,\ revision = 4)$, $rollback = (index = 7,\ revision = 5)$] $\leftarrow -$ Roll back revision 5 at index 7, leading to revision 4
ROLLBACK [$index = 9$, $revision = 5$, $change = (index = 1,\ revision = 1)$, $rollback = (index = 6,\ revision = 4)$] $\leftarrow -$ Roll back revision 4 at index 6, leading to revision 1 CHANGE [$index = 10$, $revision = 6$, $change = (index = 10,\ revision = 6)$, $rollback = (index = 1,\ revision = 1)$]

$$
\begin{array}{l}
CommitChange(n,\ i) \triangleq \\
\quad \land\ proposals[i].commit = Pending \\
\quad \land\ i - 1 \in \text{DOMAIN}\ proposals \Rightarrow \\
\qquad\quad proposals[i-1].commit \neq Pending \\
\quad \land\ configuration' = [configuration\ \text{EXCEPT}\ !.committed.index\ \ = proposals[i].change.index, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad !.committed.revision = proposals[i].change.revision, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad !.committed.values\ \ = proposals[i].change.values \mathbin{@@} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad configuration.committed.values] \\
\quad \land\ proposals' = [proposals\ \text{EXCEPT}\ ![i].commit = Complete] \\
\quad \land\ history' = Append(history,\ [\\
\qquad\qquad\qquad type\ \ \mapsto Change, \\
\qquad\qquad\qquad phase\ \ \mapsto Commit, \\
\qquad\qquad\qquad revision \mapsto proposals[i].change.revision]) \\
\quad \land\ \text{UNCHANGED}\ \langle target \rangle
\end{array}
$$

$$
\begin{array}{l}
ApplyChange(n,\ i) \triangleq \\
\quad \land\ proposals[i].apply = Pending \\
\quad \land\ proposals[i].commit = Complete \\
\quad \land\ i - 1 \in \text{DOMAIN}\ proposals \Rightarrow \\
\qquad\quad proposals[i-1].apply \neq Pending \\
\quad \land\ \lor\ \land\ i - 1 \in \text{DOMAIN}\ proposals \Rightarrow \\
\qquad\qquad\quad proposals[i-1].apply = Complete \\
\qquad\quad \land\ configuration.state = Complete \\
\qquad\quad \land\ configuration.term = mastership.term \\
\qquad\quad \land\ conn[n].id = mastership.conn \\
\qquad\quad \land\ conn[n].connected \\
\qquad\quad \land\ target.running \\
\qquad\qquad \text{Apply to the target successfully.} \\
\qquad\quad \land\ \lor\ \land\ target' = [target\ \text{EXCEPT}\ !.values = proposals[i].change.values \mathbin{@@} target.values] \\
\qquad\qquad\quad \land\ configuration' = [configuration\ \text{EXCEPT}\ !.applied.index\ \ = proposals[i].change.index, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad !.applied.revision = proposals[i].change.revision, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad !.applied.values\ \ = proposals[i].change.values \mathbin{@@} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad configuration.applied.values]
\end{array}
$$

```
                    ∧ proposals′ = [proposals EXCEPT ![i].apply = Complete]
                    ∧ history′ = Append(history, [
                                     type     ↦ Change,
                                     phase    ↦ Apply,
                                     revision ↦ proposals[i].change.revision])
                Apply to the target failed.
                ∨ ∧ proposals′ = [proposals EXCEPT ![i].apply = Failed]
                  ∧ UNCHANGED ⟨configuration, target, history⟩
            ∨ ∧ i − 1 ∈ DOMAIN proposals
              ∧ proposals[i − 1].apply ∈ {Aborted, Failed}
              ∧ proposals′ = [proposals EXCEPT ![i].apply = Aborted]
              ∧ UNCHANGED ⟨configuration, target, history⟩

ReconcileChange(n, i) ≜
    ∧ proposals[i].type = Change
    ∧ ∨ CommitChange(n, i)
      ∨ ApplyChange(n, i)

CommitRollback(n, i) ≜
    ∧ proposals[i].commit = Pending
    ∧ i − 1 ∈ DOMAIN proposals ⇒
            proposals[i − 1].commit ≠ Pending
    ∧ configuration′ = [configuration EXCEPT !.committed.index    = proposals[i].change.index,
                                             !.committed.revision = proposals[i].change.revision,
                                             !.committed.values   = proposals[i].change.values @@
                                                                    configuration.committed.values]
    ∧ proposals′ = [proposals EXCEPT ![i].commit = Complete]
    ∧ history′ = Append(history, [
                     type     ↦ Rollback,
                     phase    ↦ Commit,
                     revision ↦ proposals[i].rollback.revision])
    ∧ UNCHANGED ⟨target⟩

ApplyRollback(n, i) ≜
    ∧ proposals[i].apply = Pending
    ∧ proposals[i].commit = Complete
    ∧ i − 1 ∈ DOMAIN proposals ⇒
            proposals[i − 1].apply ≠ Pending
    ∧ ∨ ∧ proposals[proposals[i].rollback.index].apply ∈ {Complete, Failed}
          ∧ configuration.state = Complete
          ∧ configuration.term = mastership.term
          ∧ conn[n].id = mastership.conn
          ∧ conn[n].connected
          ∧ target.running
          ∧ target′ = [target EXCEPT !.values = proposals[i].change.values @@ target.values]
          ∧ configuration′ = [configuration EXCEPT !.applied.index    = proposals[i].change.index,
```

```
                                                !.applied.revision = proposals[i].change.revision,
                                                !.applied.values   = proposals[i].change.values @@
                                                                     configuration.applied.values]
        ∧ proposals' = [proposals EXCEPT ![i].apply = Complete]
        ∧ history' = Append(history, [
                                type     ↦ Rollback,
                                phase    ↦ Apply,
                                revision ↦ proposals[i].rollback.revision])
    ∨ ∧ proposals[proposals[i].rollback.index].apply = Aborted
      ∧ proposals' = [proposals EXCEPT ![i].apply = Aborted]
      ∧ UNCHANGED ⟨configuration, target, history⟩

ReconcileRollback(n, i) ≜
  ∧ proposals[i].type = Rollback
  ∧ ∨ CommitRollback(n, i)
    ∨ ApplyRollback(n, i)

ReconcileProposal(n, i) ≜
  ∧ i ∈ DOMAIN proposals
  ∧ ∨ ReconcileChange(n, i)
    ∨ ReconcileRollback(n, i)
  ∧ UNCHANGED ⟨mastership, conn⟩
```