```
──────────────────────── MODULE Config ────────────────────────
EXTENDS
    Northbound,
    Transactions,
    Proposals,
    Configurations,
    Southbound

INSTANCE Naturals

INSTANCE FiniteSets

INSTANCE Sequences

LOCAL INSTANCE TLC

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

───────────────────────────────────────────────────────────────

Formal specification, constraints, and theorems.
Init ≜
    ∧ InitTransaction
    ∧ InitProposal
    ∧ InitConfiguration
    ∧ InitNorthbound
    ∧ InitSouthbound

Next ≜
    ∨ ∧ NextTransaction
      ∧ UNCHANGED ⟨configuration, target, mastership⟩
    ∨ ∧ NextProposal
      ∧ UNCHANGED ⟨transaction⟩
    ∨ ∧ NextConfiguration
      ∧ UNCHANGED ⟨transaction, proposal⟩
    ∨ ∧ NextNorthbound
      ∧ UNCHANGED ⟨proposal, configuration, target, mastership⟩
    ∨ ∧ NextSouthbound
      ∧ UNCHANGED ⟨transaction, proposal, configuration⟩

Spec ≜ Init ∧ □[Next]_vars ∧ WF_vars(Next)

Order ≜
    ∀ t ∈ DOMAIN proposal :
      ∀ i ∈ DOMAIN proposal[t] :
        ∧ ∧ proposal[t][i].phase = ProposalCommit
          ∧ proposal[t][i].state  = ProposalInProgress
```

```
            $\Rightarrow \neg \exists j \in \text{DOMAIN } proposal[t] :$
                    $\land j > i$
                    $\land proposal[t][j].phase = ProposalCommit$
                    $\land proposal[t][j].state = ProposalComplete$
        $\land$ $\land proposal[t][i].phase = ProposalApply$
            $\land proposal[t][i].state = ProposalInProgress$
            $\Rightarrow \neg \exists j \in \text{DOMAIN } proposal[t] :$
                    $\land j > i$
                    $\land proposal[t][j].phase = ProposalApply$
                    $\land proposal[t][j].state = ProposalComplete$

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

$Isolation \triangleq$
    $\forall i \in \text{DOMAIN } transaction :$
      $\land$ $\land transaction[i].phase = TransactionCommit$
          $\land transaction[i].state = TransactionInProgress$
          $\land transaction[i].isolation = Serializable$
          $\Rightarrow \neg \exists j \in \text{DOMAIN } transaction :$
```

```
                $\land j > i$
                $\land \textsc{domain}\ transaction[j].targets \cap \textsc{domain}\ transaction[i].targets \neq \{\}$
                $\land transaction[j].phase = TransactionCommit$
        $\land$ $\land transaction[i].phase = TransactionApply$
           $\land transaction[i].state\ = TransactionInProgress$
           $\land transaction[i].isolation = Serializable$
           $\Rightarrow \neg\exists j \in \textsc{domain}\ transaction :$
                $\land j > i$
                $\land \textsc{domain}\ transaction[j].targets \cap \textsc{domain}\ transaction[i].targets \neq \{\}$
                $\land transaction[j].phase = TransactionApply$
```

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

$\textsc{theorem}\ Spec \Rightarrow Safety$

```
$Terminated(i)\ \triangleq$
    $\land i \in \textsc{domain}\ transaction$
    $\land transaction[i].phase \in \{TransactionApply,\ TransactionAbort\}$
    $\land transaction[i].state\ = TransactionComplete$
```

```
$Termination\ \triangleq$
    $\forall i \in 1 \mathrel{..} Len(transaction) : Terminated(i)$
```

$Liveness\ \triangleq\ \Diamond Termination$

$\textsc{theorem}\ Spec \Rightarrow Liveness$

* Modification History
* Last modified Sun *Feb* 20 10:08:30 *PST* 2022 by *jordanhalterman*
* Created *Wed Sep* 22 13:22:32 *PDT* 2021 by *jordanhalterman*