MODULE *Config*

EXTENDS
Northbound,
Transaction,
Proposal,
Configuration,
Southbound

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

LOCAL INSTANCE *TLC*

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

Formal specification, constraints, and theorems.

$Init \triangleq$
$\land\ InitTransaction$
$\land\ InitProposal$
$\land\ InitConfiguration$
$\land\ InitNorthbound$
$\land\ InitSouthbound$

$Next \triangleq$
$\lor\ \land\ NextTransaction$
$\quad\land$ UNCHANGED $\langle configuration, target, mastership \rangle$
$\lor\ \land\ NextProposal$
$\quad\land$ UNCHANGED $\langle transaction \rangle$
$\lor\ \land\ NextConfiguration$
$\quad\land$ UNCHANGED $\langle transaction, proposal \rangle$
$\lor\ \land\ NextNorthbound$
$\quad\land$ UNCHANGED $\langle proposal, configuration, target, mastership \rangle$
$\lor\ \land\ NextSouthbound$
$\quad\land$ UNCHANGED $\langle transaction, proposal, configuration \rangle$

$Spec \triangleq Init \land \Box[Next]_{vars} \land \mathrm{WF}_{vars}(Next)$

$Order \triangleq$
$\forall\, t \in$ DOMAIN $proposal :$
$\forall\, i \in$ DOMAIN $proposal[t] :$
$\land\ \land\ proposal[t][i].phase = ProposalCommit$
$\quad\land\ proposal[t][i].state\ = ProposalInProgress$

```
        ⇒ ¬∃ j ∈ DOMAIN proposal[t] :
                ∧ j > i
                ∧ proposal[t][j].phase = ProposalCommit
                ∧ proposal[t][j].state  = ProposalComplete
    ∧ ∧ proposal[t][i].phase = ProposalApply
      ∧ proposal[t][i].state  = ProposalInProgress
      ⇒ ¬∃ j ∈ DOMAIN proposal[t] :
                ∧ j > i
                ∧ proposal[t][j].phase = ProposalApply
                ∧ proposal[t][j].state  = ProposalComplete

Consistency ≜
  ∀ t ∈ DOMAIN target :
    LET
        Compute the transaction indexes that have been applied to the target
      targetIndexes ≜ {i ∈ DOMAIN transaction :
                          ∧ i ∈ DOMAIN proposal[t]
                          ∧ proposal[t][i].phase = ProposalApply
                          ∧ proposal[t][i].state  = ProposalComplete
                          ∧ t ∈ transaction[i].targets
                          ∧ ¬∃ j ∈ DOMAIN transaction :
                                ∧ j > i
                                ∧ transaction[j].type = TransactionRollback
                                ∧ transaction[j].rollback = i
                                ∧ transaction[j].phase = TransactionApply
                                ∧ transaction[j].state  = TransactionComplete}
        Compute the set of paths in the target that have been updated by transactions
      appliedPaths ≜ UNION {DOMAIN proposal[t][i].change.values : i ∈ targetIndexes}
        Compute the highest index applied to the target for each path
      pathIndexes ≜ [p ∈ appliedPaths ↦ CHOOSE i ∈ targetIndexes :
                          ∀ j ∈ targetIndexes :
                              ∧ i ≥ j
                              ∧ p ∈ DOMAIN proposal[t][i].change.values]
        Compute the expected target configuration based on the last indexes applied
        to the target for each path.
      expectedConfig ≜ [p ∈ DOMAIN pathIndexes ↦ proposal[t][pathIndexes[p]].change.values[p]]
    IN
      target[t] = expectedConfig

Isolation ≜
  ∀ i ∈ DOMAIN transaction :
    ∧ ∧ transaction[i].phase = TransactionCommit
      ∧ transaction[i].state  = TransactionInProgress
      ∧ transaction[i].isolation = Serializable
      ⇒ ¬∃ j ∈ DOMAIN transaction :
```

$$
\begin{array}{l}
\quad\quad\quad\quad \land\ j > i \\
\quad\quad\quad\quad \land\ transaction[j].targets \cap transaction[i].targets \neq \{\} \\
\quad\quad\quad\quad \land\ transaction[j].phase = TransactionCommit \\
\quad \land\ \land\ transaction[i].phase = TransactionApply \\
\quad\quad \land\ transaction[i].state\ \ = TransactionInProgress \\
\quad\quad \land\ transaction[i].isolation = Serializable \\
\quad\quad \Rightarrow \neg\exists\, j \in \text{DOMAIN}\ transaction : \\
\quad\quad\quad\quad \land\ j > i \\
\quad\quad\quad\quad \land\ transaction[j].targets \cap transaction[i].targets \neq \{\} \\
\quad\quad\quad\quad \land\ transaction[j].phase = TransactionApply
\end{array}
$$

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

THEOREM $Spec \Rightarrow Safety$

$$
\begin{array}{l}
Terminated(i) \triangleq \\
\quad \land\ i \in \text{DOMAIN}\ transaction \\
\quad \land\ transaction[i].phase \in \{TransactionApply,\ TransactionAbort\} \\
\quad \land\ transaction[i].state\ \ = TransactionComplete
\end{array}
$$

$$
\begin{array}{l}
Termination \triangleq \\
\quad \forall\, i \in 1\,..\,Len(transaction) : Terminated(i)
\end{array}
$$

$Liveness \triangleq \Diamond Termination$

THEOREM $Spec \Rightarrow Liveness$

* Modification History
* Last modified Sun *Feb* 20 09:10:53 *PST* 2022 by *jordanhalterman*
* Created *Wed Sep* 22 13:22:32 *PDT* 2021 by *jordanhalterman*