
MODULE *Config*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

An empty constant

CONSTANT *Nil*

Transaction type constants

CONSTANTS

TransactionChange,
TransactionRollback

Transaction isolation constants

CONSTANTS

IsolationDefault,
IsolationSerializable

Transaction status constants

CONSTANTS

TransactionInitializing,
TransactionInitialized,
TransactionValidating,
TransactionValidated,
TransactionCommitting,
TransactionCommitted,
TransactionApplying,
TransactionApplied,
TransactionFailed

TransactionStatus \triangleq

\langle *TransactionInitializing*,
TransactionInitialized,
TransactionValidating,
TransactionValidated,
TransactionCommitting,
TransactionCommitted,
TransactionApplying,
TransactionApplied,
TransactionFailed \rangle

Proposal type constants

CONSTANTS

ProposalChange,
ProposalRollback

Proposal status constants

CONSTANTS

ProposalInitializing,
ProposalInitialized,
ProposalValidating,
ProposalValidated,
ProposalCommitting,
ProposalCommitted,
ProposalApplying,
ProposalApplied,
ProposalFailed

$ProposalStatus \triangleq$
 $\langle ProposalInitializing,$
 $ProposalInitialized,$
 $ProposalValidating,$
 $ProposalValidated,$
 $ProposalCommitting,$
 $ProposalCommitted,$
 $ProposalApplying,$
 $ProposalApplied,$
 $ProposalFailed \rangle$

Configuration status constants

CONSTANTS

ConfigurationUnknown,
ConfigurationSynchronizing,
ConfigurationSynchronized,
ConfigurationPersisted,
ConfigurationFailed

CONSTANTS

Valid,
Invalid

CONSTANTS

Success,
Failure

The set of all nodes

CONSTANT *Node*

Target is the set of all targets and their possible paths and values.

Example: $Target \triangleq [$
 $\quad target1 \mapsto [\text{persistent} \mapsto \text{FALSE}, \text{values} \mapsto [$
 $\quad \quad path1 \mapsto \{ "value1", "value2" \},$
 $\quad \quad path2 \mapsto \{ "value2", "value3" \}]],$
 $\quad target2 \mapsto [\text{persistent} \mapsto \text{TRUE}, \text{values} \mapsto [$
 $\quad \quad path2 \mapsto \{ "value3", "value4" \},$
 $\quad \quad path3 \mapsto \{ "value4", "value5" \}]]]$

CONSTANT *Target*

$Phase(S, s) \triangleq \text{CHOOSE } i \in \text{DOMAIN } S : S[i] = s$

$TransactionPhase(s) \triangleq Phase(TransactionStatus, s)$

$ProposalPhase(s) \triangleq Phase(ProposalStatus, s)$

ASSUME *Nil* ∈ STRING

ASSUME *TransactionInitializing* ∈ STRING

ASSUME *TransactionInitialized* ∈ STRING

ASSUME *TransactionValidating* ∈ STRING

ASSUME *TransactionValidated* ∈ STRING

ASSUME *TransactionCommitting* ∈ STRING

ASSUME *TransactionCommitted* ∈ STRING

ASSUME *TransactionApplying* ∈ STRING

ASSUME *TransactionApplied* ∈ STRING

ASSUME *TransactionFailed* ∈ STRING

ASSUME *ProposalInitializing* ∈ STRING

ASSUME *ProposalInitialized* ∈ STRING

ASSUME *ProposalValidating* ∈ STRING

ASSUME *ProposalValidated* ∈ STRING

ASSUME *ProposalCommitting* ∈ STRING

ASSUME *ProposalCommitted* ∈ STRING

ASSUME *ProposalApplying* ∈ STRING

ASSUME *ProposalApplied* ∈ STRING

ASSUME *ProposalFailed* ∈ STRING

ASSUME *ConfigurationUnknown* ∈ STRING

ASSUME *ConfigurationSynchronizing* ∈ STRING

ASSUME *ConfigurationSynchronized* ∈ STRING

ASSUME *ConfigurationPersisted* ∈ STRING

ASSUME *ConfigurationFailed* ∈ STRING

ASSUME $\wedge IsFiniteSet(Node)$

$\wedge \forall n \in Node :$

$\wedge n \notin \text{DOMAIN } Target$

$\wedge n \in \text{STRING}$

ASSUME $\wedge \forall t \in \text{DOMAIN } Target :$
 $\wedge t \notin Node$
 $\wedge t \in \text{STRING}$
 $\wedge Target[t].persistent \in \text{BOOLEAN}$
 $\wedge \forall p \in \text{DOMAIN } Target[t].values :$
 $IsFiniteSet(Target[t].values[p])$

Configuration update/rollback requests are tracked and processed through two data types. Transactions represent the lifecycle of a single configuration change request and are stored in an append-only log. Configurations represent the desired configuration of a *gNMI* target based on the aggregate of relevant changes in the Transaction log.

```

TYPE TransactionType ::= type ∈
  { TransactionChange,
    TransactionRollback }

TYPE TransactionStatus ::= status ∈
  { TransactionInitializing,
    TransactionInitialized,
    TransactionValidating,
    TransactionValidated,
    TransactionCommitting,
    TransactionCommitted,
    TransactionApplying,
    TransactionApplied,
    TransactionFailed }

TYPE Transaction  $\triangleq$  [
  type      ::= type ∈ TransactionType,
  index     ::= index ∈ Nat,
  isolation ::= isolation ∈ { IsolationDefault, IsolationSerializable }
  values ::= [
    target ∈ SUBSET (DOMAIN Target)  $\mapsto$  [ path ∈ SUBSET (DOMAIN Target[target].values)  $\mapsto$ 
      [
        value ::= value ∈ STRING,
        delete ::= delete ∈ BOOLEAN ]]],
  rollback ::= index ∈ Nat,
  targets   ::= targets ∈ SUBSET (DOMAIN Target)
  status    ::= status ∈ TransactionStatus]

TYPE ProposalStatus ::= status ∈
  { ProposalInitializing,
    ProposalInitialized,
    ProposalValidating,
    ProposalValidated,
    ProposalCommitting,
    ProposalCommitted,
    ProposalApplying,
    ProposalApplied,
    ProposalFailed }

```

```

TYPE Proposal  $\triangleq$  [
  index      ::= index  $\in$  Nat,
  values     ::= [ path  $\in$  SUBSET (DOMAIN Target[target].values)  $\mapsto$  [
    value ::= value  $\in$  STRING,
    delete ::= delete  $\in$  BOOLEAN ]],
  rollback   ::= index  $\in$  Nat,
  prevIndex  ::= prevIndex  $\in$  Nat,
  nextIndex  ::= nextIndex  $\in$  Nat,
  rollbackIndex ::= rollbackIndex  $\in$  Nat,
  rollbackValues ::= [ path  $\in$  SUBSET (DOMAIN Target[target].values)  $\mapsto$  [
    value ::= value  $\in$  STRING,
    delete ::= delete  $\in$  BOOLEAN ]],
  status     ::= status  $\in$  ProposalStatus]

TYPE ConfigurationStatus ::= status  $\in$ 
{ ConfigurationUnknown,
  ConfigurationSynchronizing,
  ConfigurationSynchronized,
  ConfigurationPersisted,
  ConfigurationFailed}

TYPE Configuration  $\triangleq$  [
  id          ::= id  $\in$  STRING,
  target      ::= target  $\in$  STRING,
  values      ::= [ path  $\in$  SUBSET (DOMAIN Target[target])  $\mapsto$  [
    value ::= value  $\in$  STRING,
    index ::= index  $\in$  Nat,
    deleted ::= delete  $\in$  BOOLEAN ]],
  configIndex ::= configIndex  $\in$  Nat,
  configTerm  ::= configTerm  $\in$  Nat,
  proposedIndex ::= proposedIndex  $\in$  Nat,
  committedIndex ::= committedIndex  $\in$  Nat,
  appliedIndex ::= appliedIndex  $\in$  Nat,
  appliedTerm ::= appliedTerm  $\in$  Nat,
  appliedValues ::= [ path  $\in$  SUBSET (DOMAIN Target[target])  $\mapsto$  [
    value ::= value  $\in$  STRING,
    index ::= index  $\in$  Nat,
    deleted ::= delete  $\in$  BOOLEAN ]],
  status ::= status  $\in$  ConfigurationStatus]

```

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 masterhips

VARIABLE *mastership*

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

This section models *mastership* for the configuration service.

Mastership is used primarily to track the lifecycle of individual configuration targets and react to state changes on the southbound. Each target is assigned a master from the *Node* set, and masters can be unset when the target disconnects.

Set node n as the master for target t

$\text{SetMaster}(n, t) \triangleq$
 $\wedge \text{mastership}[t].\text{master} \neq n$
 $\wedge \text{mastership}' = [\text{mastership} \text{ EXCEPT } ![t].\text{term} = \text{mastership}[t].\text{term} + 1,$
 $\phantom{\wedge \text{mastership}' = } ![t].\text{master} = n]$
 $\wedge \text{UNCHANGED } \langle \text{transaction}, \text{proposal}, \text{configuration}, \text{target} \rangle$

$\text{UnsetMaster}(t) \triangleq$
 $\wedge \text{mastership}[t].\text{master} \neq \text{Nil}$
 $\wedge \text{mastership}' = [\text{mastership} \text{ EXCEPT } ![t].\text{master} = \text{Nil}]$
 $\wedge \text{UNCHANGED } \langle \text{transaction}, \text{proposal}, \text{configuration}, \text{target} \rangle$

This section models configuration changes and rollbacks. Changes are appended to the transaction log and processed asynchronously.

$\text{Value}(s, t, p) \triangleq$
 $\text{LET } \text{value} \triangleq \text{CHOOSE } v \in s : v.\text{target} = t \wedge v.\text{path} = p$
 IN
 $[\text{value} \mapsto \text{value}.\text{value},$
 $ \text{delete} \mapsto \text{value}.\text{delete}]$

$\text{Paths}(s, t) \triangleq$
 $[p \in \{v.\text{path} : v \in \{v \in s : v.\text{target} = t\}\} \mapsto \text{Value}(s, t, p)]$

$\text{Changes}(s) \triangleq$
 $[t \in \{v.\text{target} : v \in s\} \mapsto \text{Paths}(s, t)]$

$\text{ValidValues}(t, p) \triangleq$
 $\text{UNION } \{ \{[\text{value} \mapsto v, \text{delete} \mapsto \text{FALSE}] : v \in \text{Target}[t].\text{values}[p]\}, \{[\text{value} \mapsto \text{Nil}, \text{delete} \mapsto \text{TRUE}]\} \}$

$\text{ValidPaths}(t) \triangleq$
 $\text{UNION } \{ \{v @@@ [\text{path} \mapsto p] : v \in \text{ValidValues}(t, p)\} : p \in \text{DOMAIN } \text{Target}[t].\text{values} \}$

$\text{ValidTargets} \triangleq$
 $\text{UNION } \{ \{p @@@ [\text{target} \mapsto t] : p \in \text{ValidPaths}(t)\} : t \in \text{DOMAIN } \text{Target} \}$

The set of all valid sets of changes to all targets and their paths.

The set of possible changes is computed from the *Target* model value.

$$\begin{aligned}
 \text{ValidChanges} &\triangleq \\
 \text{LET } \text{changeSets} &\triangleq \{s \in \text{SUBSET } \text{ValidTargets} : \\
 &\quad \forall t \in \text{DOMAIN } \text{Target} : \\
 &\quad \quad \forall p \in \text{DOMAIN } \text{Target}[t].\text{values} : \\
 &\quad \quad \quad \text{Cardinality}(\{v \in s : v.\text{target} = t \wedge v.\text{path} = p\}) \leq 1\} \\
 \text{IN} & \\
 &\{ \text{Changes}(s) : s \in \text{changeSets} \}
 \end{aligned}$$

The next available index in the transaction log.

This is computed as the max of the existing indexes in the log to allow for changes to the log (*e.g.* log compaction) to be modeled.

$$\begin{aligned}
 \text{NextIndex} &\triangleq \\
 \text{IF DOMAIN } \text{transaction} &= \{\} \text{ THEN} \\
 &1 \\
 \text{ELSE} & \\
 \text{LET } i &\triangleq \text{CHOOSE } i \in \text{DOMAIN } \text{transaction} : \\
 &\quad \forall j \in \text{DOMAIN } \text{transaction} : i \geq j \\
 \text{IN } &i + 1
 \end{aligned}$$

Add a set of changes 'c' to the transaction log

$$\begin{aligned}
 \text{Change}(c) &\triangleq \\
 \wedge \exists \text{isolation} &\in \{\text{IsolationDefault}, \text{IsolationSerializable}\} : \\
 &\quad \wedge \text{transaction}' = \text{transaction} @@ (\text{NextIndex} :> [\text{type} \mapsto \text{TransactionChange}, \\
 &\quad \quad \quad \text{index} \mapsto \text{NextIndex}, \\
 &\quad \quad \quad \text{isolation} \mapsto \text{isolation}, \\
 &\quad \quad \quad \text{values} \mapsto c, \\
 &\quad \quad \quad \text{targets} \mapsto \{\}, \\
 &\quad \quad \quad \text{status} \mapsto \text{TransactionInitializing}]) \\
 \wedge \text{UNCHANGED } &\langle \text{proposal}, \text{configuration}, \text{mastership}, \text{target} \rangle
 \end{aligned}$$

Add a rollback of transaction 't' to the transaction log

$$\begin{aligned}
 \text{Rollback}(t) &\triangleq \\
 \wedge \exists \text{isolation} &\in \{\text{IsolationDefault}, \text{IsolationSerializable}\} : \\
 &\quad \wedge \text{transaction}' = \text{transaction} @@ (\text{NextIndex} :> [\text{type} \mapsto \text{TransactionRollback}, \\
 &\quad \quad \quad \text{index} \mapsto \text{NextIndex}, \\
 &\quad \quad \quad \text{isolation} \mapsto \text{isolation}, \\
 &\quad \quad \quad \text{rollback} \mapsto t, \\
 &\quad \quad \quad \text{targets} \mapsto \{\}, \\
 &\quad \quad \quad \text{status} \mapsto \text{TransactionInitializing}]) \\
 \wedge \text{UNCHANGED } &\langle \text{proposal}, \text{configuration}, \text{mastership}, \text{target} \rangle
 \end{aligned}$$

This section models the Transaction log reconciler.

Transactions come in two flavors : – *Change* transactions contain a set of changes to be applied to a set of *targets* – *Rollback* transactions reference a prior change transaction to be reverted to the previous state

Both types of transaction are reconciled in stages:

- * Pending - waiting for prior transactions to complete
- * Validating - validating the requested changes
- * Applying - applying the changes to target configurations
- * Complete - completed applying changes successfully
- * Failed - failed applying changes

Reconcile a transaction

$ReconcileTransaction(n, i) \triangleq$

$$\begin{aligned}
& \wedge \vee \wedge transaction[i].status = TransactionInitializing \\
& \wedge i - 1 \in \text{DOMAIN } transaction \Rightarrow \\
& \quad TransactionPhase(transaction[i - 1].status) > TransactionPhase(TransactionInitializing) \\
& \wedge \vee \wedge transaction[i].targets = \{\} \\
& \quad \wedge \vee \wedge transaction[i].type = TransactionChange \\
& \quad \quad \wedge transaction' = [transaction \text{ EXCEPT } ![i].targets = \text{DOMAIN } transaction[i].values] \\
& \quad \quad \wedge proposal' = [t \in \text{DOMAIN } proposal \mapsto \\
& \quad \quad \quad \text{IF } t \in \text{DOMAIN } transaction[i].values \text{ THEN} \\
& \quad \quad \quad \quad proposal[t] @@ (i :> [type \mapsto ProposalChange, \\
& \quad \quad \quad \quad \quad index \mapsto i, \\
& \quad \quad \quad \quad \quad values \mapsto transaction[i].values[t], \\
& \quad \quad \quad \quad \quad prevIndex \mapsto 0, \\
& \quad \quad \quad \quad \quad nextIndex \mapsto 0, \\
& \quad \quad \quad \quad \quad rollbackIndex \mapsto 0, \\
& \quad \quad \quad \quad \quad rollbackValues \mapsto \langle \rangle, \\
& \quad \quad \quad \quad \quad status \mapsto ProposalInitializing]) \\
& \quad \quad \quad \text{ELSE} \\
& \quad \quad \quad \quad proposal[t] \\
& \quad \quad \vee \wedge transaction[i].type = TransactionRollback \\
& \quad \quad \quad \wedge \vee \wedge transaction[i].rollback \in \text{DOMAIN } transaction \\
& \quad \quad \quad \quad \wedge transaction[transaction[i].rollback].type = TransactionChange \\
& \quad \quad \quad \quad \wedge transaction' = [transaction \text{ EXCEPT } ![i].targets = \\
& \quad \quad \quad \quad \quad \text{DOMAIN } transaction[transaction[i].rollback].values] \\
& \quad \quad \quad \wedge proposal' = [t \in \text{DOMAIN } proposal \mapsto \\
& \quad \quad \quad \quad \quad \text{IF } t \in \text{DOMAIN } transaction[transaction[i].rollback].values \text{ THEN} \\
& \quad \quad \quad \quad \quad \quad proposal[t] @@ (i :> [type \mapsto ProposalRollback, \\
& \quad \quad \quad \quad \quad \quad \quad index \mapsto i, \\
& \quad \quad \quad \quad \quad \quad \quad rollback \mapsto transaction[i].rollback, \\
& \quad \quad \quad \quad \quad \quad \quad prevIndex \mapsto 0, \\
& \quad \quad \quad \quad \quad \quad \quad nextIndex \mapsto 0, \\
& \quad \quad \quad \quad \quad \quad \quad rollbackIndex \mapsto 0, \\
& \quad \quad \quad \quad \quad \quad \quad rollbackValues \mapsto \langle \rangle, \\
& \quad \quad \quad \quad \quad \quad \quad status \mapsto ProposalInitializing]) \\
& \quad \quad \quad \quad \text{ELSE} \\
& \quad \quad \quad \quad \quad proposal[t] \\
& \quad \quad \quad \quad \text{ENDIF} \\
& \quad \quad \quad \text{ENDIF} \\
& \quad \quad \text{ENDIF} \\
& \text{ENDIF}
\end{aligned}$$

ELSE

$proposal[t]$

$\vee \wedge \vee \wedge transaction[i].rollback \in \text{DOMAIN } transaction$

$\wedge transaction[transaction[i].rollback].type = \text{TransactionRollback}$

$\vee transaction[i].rollback \notin \text{DOMAIN } transaction$

$\wedge transaction' = [transaction \text{ EXCEPT } ![i].status = \text{TransactionFailed}]$

$\wedge \text{UNCHANGED } \langle proposal \rangle$

$\vee \wedge transaction[i].targets \neq \{\}$

$\wedge \vee \wedge \forall t \in transaction[i].targets : proposal[t][i].status = \text{ProposalInitialized}$

$\wedge transaction' = [transaction \text{ EXCEPT } ![i].status = \text{TransactionInitialized}]$

$\wedge \text{UNCHANGED } \langle proposal \rangle$

$\vee \wedge \exists t \in transaction[i].targets : proposal[t][i].status = \text{ProposalFailed}$

$\wedge transaction' = [transaction \text{ EXCEPT } ![i].status = \text{TransactionFailed}]$

$\wedge \text{UNCHANGED } \langle proposal \rangle$

$\vee \wedge transaction[i].status = \text{TransactionInitialized}$

$\wedge \forall t \in transaction[i].targets :$

$proposal[t][i].prevIndex \neq 0 \Rightarrow$

$(transaction[proposal[t][i].prevIndex].isolation = \text{IsolationSerializable} \Rightarrow$

$\text{TransactionPhase}(transaction[proposal[t][i].prevIndex].status) \geq$

$\text{TransactionPhase}(\text{TransactionValidated}))$

$\wedge transaction' = [transaction \text{ EXCEPT } ![i].status = \text{TransactionValidating}]$

$\wedge \text{UNCHANGED } \langle proposal \rangle$

$\vee \wedge transaction[i].status = \text{TransactionValidating}$

$\wedge \vee \wedge \exists t \in transaction[i].targets :$

$\text{ProposalPhase}(proposal[t][i].status) < \text{ProposalPhase}(\text{ProposalValidating})$

$\wedge proposal' = [t \in \text{DOMAIN } proposal \mapsto$

$\text{IF } t \in transaction[i].targets \text{ THEN}$

$[proposal[t] \text{ EXCEPT } ![i].status = \text{ProposalValidating}]$

ELSE

$proposal[t]$

$\wedge \text{UNCHANGED } \langle transaction \rangle$

$\vee \wedge \forall t \in transaction[i].targets : proposal[t][i].status = \text{ProposalValidated}$

$\wedge transaction' = [transaction \text{ EXCEPT } ![i].status = \text{TransactionValidated}]$

$\wedge \text{UNCHANGED } \langle proposal \rangle$

$\vee \wedge \exists t \in transaction[i].targets : proposal[t][i].status = \text{ProposalFailed}$

$\wedge transaction' = [transaction \text{ EXCEPT } ![i].status = \text{TransactionFailed}]$

$\wedge \text{UNCHANGED } \langle proposal \rangle$

$\vee \wedge transaction[i].status = \text{TransactionValidated}$

$\wedge \forall t \in transaction[i].targets :$

$proposal[t][i].prevIndex \neq 0 \Rightarrow$

$(transaction[proposal[t][i].prevIndex].isolation = \text{IsolationSerializable} \Rightarrow$

$\text{TransactionPhase}(transaction[proposal[t][i].prevIndex].status) \geq$

$\text{TransactionPhase}(\text{TransactionCommitted}))$

$\wedge transaction' = [transaction \text{ EXCEPT } ![i].status = \text{TransactionCommitting}]$

$\wedge \text{UNCHANGED } \langle proposal \rangle$

$$\begin{aligned}
& \vee \wedge \text{transaction}[i].\text{status} = \text{TransactionCommitting} \\
& \wedge \vee \wedge \exists t \in \text{transaction}[i].\text{targets} : \\
& \quad \text{ProposalPhase}(\text{proposal}[t][i].\text{status}) < \text{ProposalPhase}(\text{ProposalCommitting}) \\
& \wedge \text{proposal}' = [t \in \text{DOMAIN } \text{proposal} \mapsto \\
& \quad \text{IF } t \in \text{transaction}[i].\text{targets} \text{ THEN} \\
& \quad \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{status} = \text{ProposalCommitting}] \\
& \quad \text{ELSE} \\
& \quad \quad \text{proposal}[t]] \\
& \wedge \text{UNCHANGED } \langle \text{transaction} \rangle \\
& \vee \wedge \forall t \in \text{transaction}[i].\text{targets} : \text{proposal}[t][i].\text{status} = \text{ProposalCommitted} \\
& \wedge \text{transaction}' = [\text{transaction} \text{ EXCEPT } ![i].\text{status} = \text{TransactionCommitted}] \\
& \wedge \text{UNCHANGED } \langle \text{proposal} \rangle \\
& \vee \wedge \exists t \in \text{transaction}[i].\text{targets} : \text{proposal}[t][i].\text{status} = \text{ProposalFailed} \\
& \wedge \text{transaction}' = [\text{transaction} \text{ EXCEPT } ![i].\text{status} = \text{TransactionFailed}] \\
& \wedge \text{UNCHANGED } \langle \text{proposal} \rangle \\
& \vee \wedge \text{transaction}[i].\text{status} = \text{TransactionCommitted} \\
& \wedge \forall t \in \text{transaction}[i].\text{targets} : \\
& \quad \text{proposal}[t][i].\text{prevIndex} \neq 0 \Rightarrow \\
& \quad (\text{transaction}[\text{proposal}[t][i].\text{prevIndex}].\text{isolation} = \text{IsolationSerializable} \Rightarrow \\
& \quad \quad \text{TransactionPhase}(\text{transaction}[\text{proposal}[t][i].\text{prevIndex}].\text{status}) \geq \\
& \quad \quad \text{TransactionPhase}(\text{TransactionApplied})) \\
& \wedge \text{transaction}' = [\text{transaction} \text{ EXCEPT } ![i].\text{status} = \text{TransactionApplying}] \\
& \wedge \text{UNCHANGED } \langle \text{proposal} \rangle \\
& \vee \wedge \text{transaction}[i].\text{status} = \text{TransactionApplying} \\
& \wedge \vee \wedge \exists t \in \text{transaction}[i].\text{targets} : \\
& \quad \text{ProposalPhase}(\text{proposal}[t][i].\text{status}) < \text{ProposalPhase}(\text{ProposalApplying}) \\
& \wedge \text{proposal}' = [t \in \text{DOMAIN } \text{proposal} \mapsto \\
& \quad \text{IF } t \in \text{transaction}[i].\text{targets} \text{ THEN} \\
& \quad \quad [\text{proposal}[t] \text{ EXCEPT } ![i].\text{status} = \text{ProposalApplying}] \\
& \quad \text{ELSE} \\
& \quad \quad \text{proposal}[t]] \\
& \wedge \text{UNCHANGED } \langle \text{transaction} \rangle \\
& \vee \wedge \forall t \in \text{transaction}[i].\text{targets} : \text{proposal}[t][i].\text{status} = \text{ProposalApplied} \\
& \wedge \text{transaction}' = [\text{transaction} \text{ EXCEPT } ![i].\text{status} = \text{TransactionApplied}] \\
& \wedge \text{UNCHANGED } \langle \text{proposal} \rangle \\
& \vee \wedge \exists t \in \text{transaction}[i].\text{targets} : \text{proposal}[t][i].\text{status} = \text{ProposalFailed} \\
& \wedge \text{transaction}' = [\text{transaction} \text{ EXCEPT } ![i].\text{status} = \text{TransactionFailed}] \\
& \wedge \text{UNCHANGED } \langle \text{proposal} \rangle \\
& \vee \wedge \text{transaction}[i].\text{status} = \text{TransactionApplied} \\
& \wedge \text{UNCHANGED } \langle \text{configuration}, \text{mastership}, \text{target} \rangle
\end{aligned}$$

Reconcile a proposal

$$\begin{aligned}
& \text{ReconcileProposal}(n, t, i) \triangleq \\
& \wedge \vee \wedge \text{proposal}[t][i].\text{status} = \text{ProposalInitializing} \\
& \wedge \vee \wedge \text{configuration}[t].\text{proposedIndex} > 0
\end{aligned}$$

$$\begin{aligned}
& \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\text{proposal}[t] \text{ EXCEPT} \\
& \quad ![i] = [\text{status} \mapsto \text{ProposalInitialized}, \\
& \quad \quad \text{prevIndex} \mapsto \text{configuration}[t].\text{proposedIndex}] @@ \text{proposal}[t][i], \\
& \quad ![\text{configuration}[t].\text{proposedIndex}] = [\text{nextIndex} \mapsto i] @@ \\
& \quad \quad \text{proposal}[t][\text{configuration}[t].\text{proposedIndex}]] \\
& \vee \wedge \text{configuration}[t].\text{proposedIndex} = 0 \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\text{proposal}[t] \text{ EXCEPT } ![i].\text{status} = \text{ProposalInitialized}]] \\
& \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{proposedIndex} = i] \\
& \wedge \text{UNCHANGED } \langle \text{target} \rangle \\
& \vee \wedge \text{proposal}[t][i].\text{status} = \text{ProposalValidating} \\
& \wedge \text{configuration}[t].\text{committedIndex} = \text{proposal}[t][i].\text{prevIndex} \\
& \wedge \vee \wedge \text{proposal}[t][i].\text{type} = \text{ProposalChange} \\
& \quad \wedge \text{LET } \text{rollbackIndex} \triangleq \text{configuration}[t].\text{configIndex} \\
& \quad \quad \text{rollbackValues} \triangleq [p \in \text{DOMAIN } \text{proposal}[t][i].\text{values} \mapsto [\\
& \quad \quad \quad p \mapsto \text{IF } p \in \text{DOMAIN } \text{configuration}[t].\text{values} \text{ THEN} \\
& \quad \quad \quad \quad \text{configuration}[t].\text{values}[p] \\
& \quad \quad \quad \text{ELSE} \\
& \quad \quad \quad \quad [\text{delete} \mapsto \text{TRUE}]]] \\
& \text{IN } \exists r \in \{ \text{Valid}, \text{Invalid} \} : \\
& \quad \vee \wedge r = \text{Valid} \\
& \quad \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\\
& \quad \quad \quad \text{proposal}[t] \text{ EXCEPT } ![i].\text{rollbackIndex} = \text{rollbackIndex}, \\
& \quad \quad \quad \quad ![i].\text{rollbackValues} = \text{rollbackValues}, \\
& \quad \quad \quad \quad \quad ![i].\text{status} = \text{ProposalValidated}]] \\
& \quad \quad \wedge \text{UNCHANGED } \langle \text{configuration} \rangle \\
& \quad \vee \wedge r = \text{Invalid} \\
& \quad \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{committedIndex} = i] \\
& \quad \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\\
& \quad \quad \quad \text{proposal}[t] \text{ EXCEPT } ![i].\text{status} = \text{ProposalFailed}]] \\
& \vee \wedge \text{proposal}[t][i].\text{type} = \text{ProposalRollback} \\
& \quad \wedge \vee \wedge \text{configuration}[t].\text{index} = \text{proposal}[t][i].\text{rollback} \\
& \quad \quad \wedge \vee \wedge \text{proposal}[t][\text{proposal}[t][i].\text{rollback}].\text{type} = \text{ProposalChange} \\
& \quad \quad \quad \wedge \text{LET } \text{rollbackIndex} \triangleq \text{proposal}[t][\text{proposal}[t][i].\text{rollback}].\text{rollbackIndex} \\
& \quad \quad \quad \quad \text{rollbackValues} \triangleq \text{proposal}[t][\text{proposal}[t][i].\text{rollback}].\text{rollbackValues} \\
& \quad \quad \text{IN } \exists r \in \{ \text{Valid}, \text{Invalid} \} : \\
& \quad \quad \quad \vee \wedge r = \text{Valid} \\
& \quad \quad \quad \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\\
& \quad \quad \quad \quad \quad \text{proposal}[t] \text{ EXCEPT } ![i].\text{rollbackIndex} = \text{rollbackIndex}, \\
& \quad \quad \quad \quad \quad \quad ![i].\text{rollbackValues} = \text{rollbackValues}, \\
& \quad \quad \quad \quad \quad \quad \quad ![i].\text{status} = \text{ProposalValidated}]] \\
& \quad \quad \quad \quad \wedge \text{UNCHANGED } \langle \text{configuration} \rangle \\
& \quad \quad \vee \wedge r = \text{Invalid} \\
& \quad \quad \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{committedIndex} = i] \\
& \quad \quad \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![t] = [\\
& \quad \quad \quad \quad \text{proposal}[t] \text{ EXCEPT } ![i].\text{status} = \text{ProposalFailed}]]
\end{aligned}$$

$$\begin{aligned}
& \wedge \text{mastership}[t].\text{term} > \text{configuration}[t].\text{configTerm} \\
& \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{configTerm} = \text{mastership}[t].\text{term}, \\
& \hspace{15em} ![t].\text{status} = \text{ConfigurationSynchronizing}] \\
& \wedge \text{UNCHANGED } \langle \text{target} \rangle \\
\vee & \wedge \neg \text{Target}[t].\text{persistent} \\
& \wedge \text{configuration}[t].\text{status} \neq \text{ConfigurationUnknown} \\
& \wedge \text{mastership}[t].\text{term} = \text{configuration}[t].\text{configTerm} \\
& \wedge \text{mastership}[t].\text{master} = \text{Nil} \\
& \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{status} = \text{ConfigurationUnknown}] \\
& \wedge \text{UNCHANGED } \langle \text{target} \rangle \\
\vee & \wedge \text{configuration}[t].\text{status} = \text{ConfigurationSynchronizing} \\
& \wedge \text{mastership}[t].\text{term} = \text{configuration}[t].\text{configTerm} \\
& \wedge \text{mastership}[t].\text{master} = n \\
& \wedge \text{target}' = [\text{target} \text{ EXCEPT } ![t] = \text{configuration}[t].\text{values}] \\
& \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } ![t].\text{appliedTerm} = \text{mastership}[t].\text{term}, \\
& \hspace{15em} ![t].\text{status} = \text{ConfigurationSynchronized}] \\
& \wedge \text{UNCHANGED } \langle \text{proposal}, \text{transaction}, \text{mastership} \rangle
\end{aligned}$$

Init and next state predicates

$$\begin{aligned}
\text{Init} & \triangleq \\
& \wedge \text{transaction} = \langle \rangle \\
& \wedge \text{proposal} = [t \in \text{DOMAIN } \text{Target} \mapsto \\
& \hspace{10em} [p \in \{\} \mapsto [\text{status} \mapsto \text{ProposalInitializing}]]] \\
& \wedge \text{configuration} = [t \in \text{DOMAIN } \text{Target} \mapsto \\
& \hspace{10em} [\text{target} \mapsto t, \\
& \hspace{10em} \text{status} \mapsto \text{ConfigurationUnknown}, \\
& \hspace{10em} \text{values} \mapsto \\
& \hspace{10em} [\text{path} \in \{\} \mapsto \\
& \hspace{10em} [\text{path} \mapsto \text{path}, \\
& \hspace{10em} \text{value} \mapsto \text{Nil}, \\
& \hspace{10em} \text{index} \mapsto 0, \\
& \hspace{10em} \text{deleted} \mapsto \text{FALSE}], \\
& \hspace{10em} \text{configIndex} \mapsto 0, \\
& \hspace{10em} \text{configTerm} \mapsto 0, \\
& \hspace{10em} \text{proposedIndex} \mapsto 0, \\
& \hspace{10em} \text{committedIndex} \mapsto 0, \\
& \hspace{10em} \text{appliedIndex} \mapsto 0, \\
& \hspace{10em} \text{appliedTerm} \mapsto 0, \\
& \hspace{10em} \text{appliedValues} \mapsto \\
& \hspace{10em} [\text{path} \in \{\} \mapsto \\
& \hspace{10em} [\text{path} \mapsto \text{path}, \\
& \hspace{10em} \text{value} \mapsto \text{Nil}, \\
& \hspace{10em} \text{index} \mapsto 0,
\end{aligned}$$

$$\begin{aligned}
& \text{deleted} \mapsto \text{FALSE}]]]] \\
\wedge \text{target} &= [t \in \text{DOMAIN } \text{Target} \mapsto \\
& \quad [path \in \{\} \mapsto \\
& \quad \quad [value \mapsto \text{Nil}]]] \\
\wedge \text{mastership} &= [t \in \text{DOMAIN } \text{Target} \mapsto [master \mapsto \text{Nil}, term \mapsto 0]] \\
\text{Next} &\triangleq \\
& \quad \vee \exists c \in \text{ValidChanges} : \\
& \quad \quad \text{Change}(c) \\
& \quad \vee \exists t \in \text{DOMAIN } \text{transaction} : \\
& \quad \quad \text{Rollback}(t) \\
& \quad \vee \exists n \in \text{Node} : \\
& \quad \quad \exists t \in \text{DOMAIN } \text{Target} : \\
& \quad \quad \quad \text{SetMaster}(n, t) \\
& \quad \vee \exists t \in \text{DOMAIN } \text{Target} : \\
& \quad \quad \quad \text{UnsetMaster}(t) \\
& \quad \vee \exists n \in \text{Node} : \\
& \quad \quad \exists t \in \text{DOMAIN } \text{transaction} : \\
& \quad \quad \quad \text{ReconcileTransaction}(n, t) \\
& \quad \vee \exists n \in \text{Node} : \\
& \quad \quad \exists t \in \text{DOMAIN } \text{proposal} : \\
& \quad \quad \quad \exists i \in \text{DOMAIN } \text{proposal}[t] : \\
& \quad \quad \quad \quad \text{ReconcileProposal}(n, t, i) \\
& \quad \vee \exists n \in \text{Node} : \\
& \quad \quad \exists c \in \text{DOMAIN } \text{configuration} : \\
& \quad \quad \quad \text{ReconcileConfiguration}(n, c) \\
\text{Spec} &\triangleq \text{Init} \wedge \Box[\text{Next}]_{\text{vars}} \\
\text{Order} &\triangleq \text{TRUE} \quad \text{TODO redefine order spec} \\
\text{THEOREM } \text{Safety} &\triangleq \text{Spec} \Rightarrow \Box \text{Order} \\
\text{Completion} &\triangleq \forall i \in \text{DOMAIN } \text{transaction} : \\
& \quad \text{transaction}[i].\text{status} \in \{\text{TransactionApplied}, \text{TransactionFailed}\} \\
\text{THEOREM } \text{Liveness} &\triangleq \text{Spec} \Rightarrow \Diamond \text{Completion}
\end{aligned}$$

\ * Modification History
\ * Last modified Sun Feb 06 02:40:51 PST 2022 by jordanhalterman
\ * Created Wed Sep 22 13:22:32 PDT 2021 by jordanhalterman