---- MODULE *Transaction* ----

INSTANCE *Naturals*

INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

Transaction type constants
CONSTANTS
    Change,
    Rollback

Transaction isolation constants
CONSTANTS
    ReadCommitted,
    Serializable

Phase constants
CONSTANTS
    Initialize,
    Validate,
    Abort,
    Commit,
    Apply

Status constants
CONSTANTS
    InProgress,
    Complete,
    Failed

State constants
CONSTANTS
    Pending,
    Validated,
    Committed,
    Applied,
    Aborted

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*

LOCAL $InitState \triangleq$
    $[transactions \mapsto transaction,$
     $proposals \quad \mapsto [t \in \text{DOMAIN } proposal \mapsto proposal[t]]]$

LOCAL $NextState \triangleq$
    $[transactions \mapsto transaction',$
     $proposals \quad \mapsto proposal']$

LOCAL $Trace \triangleq$ INSTANCE $Trace$ WITH
    $Module \quad \leftarrow$ "Transaction",
    $InitState \leftarrow InitState,$
    $NextState \leftarrow NextState$

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

Transacations proceed through a series of phases:
* *Initialize* - create and link Proposals
* *Validate* - validate changes and rollbacks
* *Commit* - commit changes to Configurations
* *Apply* - commit changes to Targets

Reconcile a transaction
$Reconcile(i) \triangleq$

   Initialize is the only transaction phase that's globally serialized. While in the Initializing phase, the reconciler checks whether the prior transaction has been Initialized before creating Proposals in the *Initialize* phase. Once all of the transaction's proposals have been Initialized, the transaction will be marked Initialized. If any proposal is *Failed*, the transaction will be marked *Failed* as well.

  $\land \lor \land transaction[i].phase = Initialize$
    $\land \lor \land transaction[i].state = InProgress$

      All prior transaction must be initialized before proceeding to initialize this transaction.

      $\land \lnot \exists j \in \text{DOMAIN } transaction :$
        $\land j < i$
        $\land transaction[j].phase = Initialize$
        $\land transaction[j].state \; = InProgress$

      If the transaction's targets are not yet set, create proposals and add targets to the transaction state.

      $\land \lor \land transaction[i].targets = \{\}$

        If the transaction is a change, the targets are taken

2

from the change values.
$\land \lor \land transaction[i].type = Change$
$\quad \land transaction' = [transaction \text{ EXCEPT } ![i].targets = \text{DOMAIN } transaction[i].change]$
$\quad \land proposal' = [t \in \text{DOMAIN } proposal \mapsto$
$\qquad \text{IF } t \in \text{DOMAIN } transaction[i].change \text{ THEN}$
$\qquad\quad proposal[t] @@ (i :> [type \quad \mapsto Change,$
$\qquad\qquad\qquad\qquad\qquad change \quad \mapsto$
$\qquad\qquad\qquad\qquad\qquad\quad [index \;\mapsto i,$
$\qquad\qquad\qquad\qquad\qquad\quad\; values \mapsto transaction[i].change[t]],$
$\qquad\qquad\qquad\qquad\qquad rollback \quad \mapsto$
$\qquad\qquad\qquad\qquad\qquad\quad [index \;\mapsto 0],$
$\qquad\qquad\qquad\qquad\qquad dependency \mapsto [index \mapsto 0],$
$\qquad\qquad\qquad\qquad\qquad phase \qquad \mapsto Initialize,$
$\qquad\qquad\qquad\qquad\qquad state \qquad \mapsto InProgress])$
$\qquad \text{ELSE}$
$\qquad\quad proposal[t]]$

If the transaction is a rollback, the targets affected are
the targets of the change transaction being rolled back.
$\lor \land transaction[i].type = Rollback$

If the rollback index is a valid *Change* transaction,
initialize proposals for all of the *Change* targets.
$\quad \land \lor \land transaction[i].rollback \in \text{DOMAIN } transaction$
$\qquad\quad \land transaction[transaction[i].rollback].type = Change$
$\qquad\quad \land transaction' = [transaction \text{ EXCEPT } ![i].targets =$
$\qquad\qquad\qquad\qquad\qquad \text{DOMAIN } transaction[transaction[i].rollback].change]$
$\qquad\quad \land proposal' = [t \in \text{DOMAIN } proposal \mapsto$
$\qquad\qquad \text{IF } t \in \text{DOMAIN } transaction[transaction[i].rollback].change \text{ THEN}$
$\qquad\qquad\quad proposal[t] @@ (i :> [type \qquad \mapsto Rollback,$
$\qquad\qquad\qquad\qquad\qquad\quad change \quad \mapsto$
$\qquad\qquad\qquad\qquad\qquad\quad\; [index \;\mapsto 0],$
$\qquad\qquad\qquad\qquad\qquad\quad rollback \quad \mapsto$
$\qquad\qquad\qquad\qquad\qquad\quad\; [index \;\mapsto transaction[i].rollback],$
$\qquad\qquad\qquad\qquad\qquad\quad dependency \mapsto [index \mapsto 0],$
$\qquad\qquad\qquad\qquad\qquad\quad phase \qquad \mapsto Initialize,$
$\qquad\qquad\qquad\qquad\qquad\quad state \qquad \mapsto InProgress])$
$\qquad\qquad \text{ELSE}$
$\qquad\qquad\quad proposal[t]]$

If the rollback index is not a valid *Change* transaction
fail the *Rollback* transaction.
$\quad\; \lor \land \lor \land transaction[i].rollback \in \text{DOMAIN } transaction$
$\qquad\qquad\quad \land transaction[transaction[i].rollback].type = Rollback$
$\qquad\qquad \lor transaction[i].rollback \notin \text{DOMAIN } transaction$
$\qquad\quad \land transaction' = [transaction \text{ EXCEPT } ![i].state = Failed]$
$\qquad\quad \land \text{UNCHANGED } \langle proposal \rangle$

If the transaction's proposals have been initialized, check proposals

3

for completion or failures.

$\lor \land transaction[i].targets \neq \{\}$

$\quad \land \lor \land \forall\, t \in transaction[i].targets :$
$\qquad\qquad \land proposal[t][i].phase = Initialize$
$\qquad\qquad \land proposal[t][i].state \;= Complete$
$\qquad\quad \land transaction' = [transaction \text{ EXCEPT } ![i].state = Complete]$
$\qquad\quad \land \text{UNCHANGED } \langle proposal \rangle$

$\qquad \lor \land \exists\, t \in transaction[i].targets :$
$\qquad\qquad \land proposal[t][i].phase \;= Initialize$
$\qquad\qquad \land proposal[t][i].state = Failed$
$\qquad\quad \land transaction' = [transaction \text{ EXCEPT } ![i].state = Failed]$
$\qquad\quad \land \text{UNCHANGED } \langle proposal \rangle$

$\lor \land transaction[i].state = Complete$
$\quad \land \forall\, t \in transaction[i].targets :$
$\qquad \land proposal[t][i].dependency.index \in \text{DOMAIN } transaction$
$\qquad \land transaction[proposal[t][i].dependency.index].isolation = Serializable$
$\qquad \Rightarrow transaction[proposal[t][i].dependency.index].status \in \{\, Validated, Committed, Applied, A$
$\quad \land transaction' = [transaction \text{ EXCEPT } ![i].phase = Validate,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![i].state \;= InProgress]$

$\quad \land \text{UNCHANGED } \langle proposal \rangle$

$\lor \land transaction[i].state = Failed$
$\quad \land transaction' = [transaction \text{ EXCEPT } ![i].phase = Abort,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].state \;= InProgress]$

$\quad \land \text{UNCHANGED } \langle proposal \rangle$

$\lor \land transaction[i].phase = Validate$
$\quad \land \lor \land transaction[i].state = InProgress$

$\qquad \land \lor \land \exists\, t \in transaction[i].targets :$
$\qquad\qquad\quad \land proposal[t][i].phase \neq Validate$
$\qquad\qquad\quad \land proposal' = [proposal \text{ EXCEPT } ![t] =$
$\qquad\qquad\qquad\qquad\qquad [proposal[t] \text{ EXCEPT } ![i].phase = Validate,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].state \;= InProgress]]$
$\qquad\quad \land \text{UNCHANGED } \langle transaction \rangle$

$\qquad \lor \land \forall\, t \in transaction[i].targets :$
$\qquad\qquad \land proposal[t][i].phase \;= Validate$
$\qquad\qquad \land proposal[t][i].state = Complete$

$\wedge\ transaction' = [transaction\ \text{EXCEPT}\ ![i].state\ \ = Complete,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].status = Validated]$
$\wedge\ \text{UNCHANGED}\ \langle proposal \rangle$

If any proposal has been *Failed*, mark the transaction *Failed*.
$\vee\ \wedge\ \exists\, t \in transaction[i].targets :$
$\qquad \wedge\ proposal[t][i].phase\ \ = Validate$
$\qquad \wedge\ proposal[t][i].state = Failed$
$\quad \wedge\ transaction' = [transaction\ \text{EXCEPT}\ ![i].state = Failed]$
$\quad \wedge\ \text{UNCHANGED}\ \langle proposal \rangle$

Once the transaction has been *Validated*, proceed to the *Commit* phase.
If any of the transaction's proposals depend on a *Serializable* transaction,
verify the dependency has been *Committed* to preserve serializability before
moving the transaction to the *Commit* phase.
$\vee\ \wedge\ transaction[i].state = Complete$
$\quad \wedge\ \forall\, t \in transaction[i].targets :$
$\qquad \wedge\ proposal[t][i].dependency.index \in \text{DOMAIN}\ transaction$
$\qquad \wedge\ transaction[proposal[t][i].dependency.index].isolation = Serializable$
$\qquad \Rightarrow transaction[proposal[t][i].dependency.index].status \in \{Committed,\ Applied,\ Aborted\}$
$\quad \wedge\ transaction' = [transaction\ \text{EXCEPT}\ ![i].phase = Commit,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].state\ \ = InProgress]$
$\quad \wedge\ \text{UNCHANGED}\ \langle proposal \rangle$

If the transaction failed validation, proceed to the *Abort* phase
to ensure indexes are still updated for the target configurations.
$\vee\ \wedge\ transaction[i].state = Failed$
$\quad \wedge\ transaction' = [transaction\ \text{EXCEPT}\ ![i].phase = Abort,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].state\ \ = InProgress]$
$\quad \wedge\ \text{UNCHANGED}\ \langle proposal \rangle$
$\vee\ \wedge\ transaction[i].phase = Commit$
$\quad \wedge\ \vee\ \wedge\ transaction[i].state = InProgress$

Move the transaction's proposals to the Committing state
$\qquad \wedge\ \vee\ \wedge\ \exists\, t \in transaction[i].targets :$
$\qquad\qquad \wedge\ proposal[t][i].phase \neq Commit$
$\qquad\qquad \wedge\ proposal' = [proposal\ \text{EXCEPT}\ ![t] =$
$\qquad\qquad\qquad\qquad [proposal[t]\ \text{EXCEPT}\ ![i].phase = Commit,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].state\ \ = InProgress]]$
$\qquad\quad \wedge\ \text{UNCHANGED}\ \langle transaction \rangle$

If all proposals have been *Complete*, mark the transaction *Complete*.
$\qquad \vee\ \wedge\ \forall\, t \in transaction[i].targets :$
$\qquad\qquad \wedge\ proposal[t][i].phase\ \ = Commit$
$\qquad\qquad \wedge\ proposal[t][i].state = Complete$
$\qquad\quad \wedge\ transaction' = [transaction\ \text{EXCEPT}\ ![i].state\ \ = Complete,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].status = Committed]$
$\qquad\quad \wedge\ \text{UNCHANGED}\ \langle proposal \rangle$

Once the transaction has been *Committed*, proceed to the *Apply* phase.
If any of the transaction's proposals depend on a *Serializable* transaction,

5

$\lor\ \land\ transaction[i].state = Complete$
$\qquad \land\ \forall\, t \in transaction[i].targets :$
$\qquad\qquad \land\ proposal[t][i].dependency.index \in \text{DOMAIN}\ transaction$
$\qquad\qquad \land\ transaction[proposal[t][i].dependency.index].isolation = Serializable$
$\qquad\qquad \Rightarrow transaction[proposal[t][i].dependency.index].status \in \{Applied,\ Aborted\}$
$\qquad \land\ transaction' = [transaction\ \text{EXCEPT}\ ![i].phase = Apply,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![i].state\ = InProgress]$

$\qquad \land\ \text{UNCHANGED}\ \langle proposal \rangle$

$\lor\ \land\ transaction[i].phase = Apply$
$\quad \land\ transaction[i].state\ = InProgress$

$\quad \land\ \lor\ \land\ \exists\, t \in transaction[i].targets :$
$\qquad\qquad\quad \land\ proposal[t][i].phase \neq Apply$
$\qquad\qquad\quad \land\ proposal' = [proposal\ \text{EXCEPT}\ ![t] =$
$\qquad\qquad\qquad\qquad\qquad\qquad [proposal[t]\ \text{EXCEPT}\ ![i].phase = Apply,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].state\ = InProgress]]$

$\qquad\quad \land\ \text{UNCHANGED}\ \langle transaction \rangle$

$\qquad \lor\ \land\ \forall\, t \in transaction[i].targets :$
$\qquad\qquad\quad \land\ proposal[t][i].phase\ = Apply$
$\qquad\qquad\quad \land\ proposal[t][i].state = Complete$
$\qquad\quad \land\ transaction' = [transaction\ \text{EXCEPT}\ ![i].state\ = Complete,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![i].status = Applied]$

$\qquad\quad \land\ \text{UNCHANGED}\ \langle proposal \rangle$

$\qquad \lor\ \land\ \exists\, t \in transaction[i].targets :$
$\qquad\qquad\quad \land\ proposal[t][i].phase\ = Apply$
$\qquad\qquad\quad \land\ proposal[t][i].state = Failed$
$\qquad\quad \land\ transaction' = [transaction\ \text{EXCEPT}\ ![i].state = Failed]$
$\qquad\quad \land\ \text{UNCHANGED}\ \langle proposal \rangle$

$\lor\ \land\ transaction[i].phase = Abort$
$\quad \land\ transaction[i].state\ = InProgress$

$\quad \land\ \lor\ \land\ \exists\, t \in transaction[i].targets :$
$\qquad\qquad\quad \land\ proposal[t][i].phase \neq Abort$
$\qquad\qquad\quad \land\ proposal' = [proposal\ \text{EXCEPT}\ ![t] =$
$\qquad\qquad\qquad\qquad\qquad\qquad [proposal[t]\ \text{EXCEPT}\ ![i].phase = Abort,$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].state\ = InProgress]]$

$\qquad\quad \land\ \text{UNCHANGED}\ \langle transaction \rangle$

$\qquad \lor\ \land\ \forall\, t \in transaction[i].targets :$

$$\begin{aligned}
&\wedge\ proposal[t][i].phase\ = Abort\\
&\wedge\ proposal[t][i].state = Complete\\
&\wedge\ transaction' = [transaction\ \text{EXCEPT}\ ![i].state\ = Complete,\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![i].status = Aborted]\\
&\wedge\ \text{UNCHANGED}\ \langle proposal\rangle
\end{aligned}$$

Formal specification, constraints, and theorems.

$Init\ \triangleq$
$\quad \wedge\ transaction = [i \in \{\} \mapsto$
$\qquad\qquad\qquad\qquad [type\quad \mapsto Change,$
$\qquad\qquad\qquad\qquad\ phase \mapsto Initialize,$
$\qquad\qquad\qquad\qquad\ state\ \mapsto InProgress,$
$\qquad\qquad\qquad\qquad\ status \mapsto Pending]]$
$\quad \wedge\ Trace\,!\,Init$

$Next\ \triangleq$
$\quad \vee\ \exists\, i \in \text{DOMAIN}\ transaction :$
$\qquad Trace\,!\,Step(\text{``Reconcile''},\ Reconcile(i),\ [index \mapsto i])$