MODULE *Transaction*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

An empty constant
CONSTANT *Nil*

Transaction phase constants
CONSTANTS
Change,
Rollback

Transaction phase constants
CONSTANTS
Commit,
Apply

Status constants
CONSTANTS
Pending,
InProgress,
Complete,
Aborted,
Canceled,
Failed

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

$Done \triangleq \{Complete, Aborted, Canceled, Failed\}$

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,
conns,
target

A transaction log. Transactions may either request a set
of changes to a set of targets or rollback a prior change.
VARIABLE *transactions*

A history of transaction change/rollback commit/apply events used for model checking.
VARIABLE *history*

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

$$
\begin{array}{l}
\text{LOCAL } \mathit{State} \triangleq [\\
\quad \mathit{transactions} \mapsto \mathit{transactions}, \\
\quad \mathit{configuration} \mapsto \mathit{configuration}, \\
\quad \mathit{mastership} \mapsto \mathit{mastership}, \\
\quad \mathit{conns} \mapsto \mathit{conns}, \\
\quad \mathit{target} \mapsto \mathit{target}]
\end{array}
$$

$$
\begin{array}{l}
\text{LOCAL } \mathit{Transitions} \triangleq \\
\quad \text{LET} \\
\qquad \mathit{indexes} \triangleq \{i \in \text{DOMAIN } \mathit{transactions}' : \\
\qquad\qquad\qquad i \in \text{DOMAIN } \mathit{transactions} \Rightarrow \mathit{transactions}'[i] \neq \mathit{transactions}[i]\} \\
\quad \text{IN} \quad [\mathit{transactions} \mapsto [i \in \mathit{indexes} \mapsto \mathit{transactions}'[i]]]\ @@ \\
\qquad (\text{IF } \mathit{configuration}' \neq \mathit{configuration} \text{ THEN } [\mathit{configuration} \mapsto \mathit{configuration}'] \text{ ELSE } \mathit{Empty})\ @@ \\
\qquad (\text{IF } \mathit{target}' \neq \mathit{target} \text{ THEN } [\mathit{target} \mapsto \mathit{target}'] \text{ ELSE } \mathit{Empty})\ @@ \\
\qquad (\text{IF } \mathit{Len}(\mathit{history}') > \mathit{Len}(\mathit{history}) \text{ THEN } [\mathit{event} \mapsto \mathit{history}'[\mathit{Len}(\mathit{history}')]] \text{ ELSE } \mathit{Empty})
\end{array}
$$

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

Add a change for revision 'i' to the transaction log

$$
\begin{array}{l}
AppendChange(i) \triangleq \\
\quad \land Len(transactions) = i - 1 \\
\quad \land \exists\, p \in Path,\, v \in Value : \\
\qquad \text{LET } transaction \triangleq [\\
\qquad\qquad index \mapsto Len(transactions) + 1, \\
\qquad\qquad phase \mapsto Change, \\
\qquad\qquad change \mapsto [\\
\qquad\qquad\quad index \mapsto Len(transactions) + 1, \\
\qquad\qquad\quad ordinal \mapsto 0, \\
\qquad\qquad\quad values \mapsto (p :> v), \\
\qquad\qquad\quad commit \mapsto Pending, \\
\qquad\qquad\quad apply \mapsto Pending], \\
\qquad\qquad rollback \mapsto [\\
\qquad\qquad\quad index \mapsto 0, \\
\qquad\qquad\quad ordinal \mapsto 0, \\
\qquad\qquad\quad values \mapsto Empty, \\
\qquad\qquad\quad commit \mapsto Nil, \\
\qquad\qquad\quad apply \mapsto Nil]] \\
\qquad \text{IN} \quad \land transactions' = Append(transactions, transaction) \\
\quad \land \text{UNCHANGED } \langle configuration, mastership, conns, target, history \rangle
\end{array}
$$

Add a rollback of revision 'i' to the transaction log

$$
\begin{array}{l}
RollbackChange(i) \triangleq \\
\quad \land i \in \text{DOMAIN } transactions \\
\quad \land transactions[i].phase = Change \\
\quad \land transactions[i].change.commit = Complete \\
\quad \land transactions' = [transactions \text{ EXCEPT } ![i].phase = Rollback, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![i].rollback.commit = Pending, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ![i].rollback.apply = Pending] \\
\quad \land \text{UNCHANGED } \langle configuration, mastership, conns, target, history \rangle
\end{array}
$$

$$
\begin{array}{l}
CommitChange(n, i) \triangleq \\
\quad \land \lor \land transactions[i].change.commit = Pending \\
\qquad\quad \land configuration.committed.change = i - 1 \\
\qquad\quad \land \lor \land configuration.committed.target \neq i \\
\qquad\qquad\quad \land configuration.committed.index = configuration.committed.target \\
\qquad\qquad\quad \land configuration.committed.index \in \text{DOMAIN } transactions \Rightarrow \\
\qquad\qquad\qquad\quad \lor \land configuration.committed.target = configuration.committed.index \\
\qquad\qquad\qquad\qquad \land transactions[configuration.committed.index].change.commit \in Done \\
\qquad\qquad\qquad\quad \lor \land configuration.committed.target < configuration.committed.index \\
\qquad\qquad\qquad\qquad \land transactions[configuration.committed.index].rollback.commit \in Done
\end{array}
$$

$$
\begin{array}{l}
\land\ configuration' = [configuration\ \text{EXCEPT}\ !.committed.target = i] \\
\land\ history' = Append(history, [\\
\qquad\qquad phase\ \mapsto Change, \\
\qquad\qquad event\ \mapsto Commit, \\
\qquad\qquad index\ \mapsto i, \\
\qquad\qquad status\ \mapsto InProgress]) \\
\land\ \lor\ transactions' = [transactions\ \text{EXCEPT}\ ![i].change.commit\ = InProgress, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].rollback.index\ = configuration.committed.revis \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].rollback.values\ = [\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad p \in \text{DOMAIN}\ transactions[i].change.values \mapsto \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{IF}\ p \in \text{DOMAIN}\ configuration.committed.values \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad configuration.committed.values[p] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE}\ Nil]] \\
\qquad \lor\ \text{UNCHANGED}\ \langle transactions \rangle \\
\lor\ \land\ configuration.committed.target = i \\
\qquad \land\ transactions' = [transactions\ \text{EXCEPT}\ ![i].change.commit\ = InProgress, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].rollback.index\ = configuration.committed.revisio \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].rollback.values\ = [\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad p \in \text{DOMAIN}\ transactions[i].change.values \mapsto \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{IF}\ p \in \text{DOMAIN}\ configuration.committed.values\ \text{TH} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad configuration.committed.values[p] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE}\ Nil]] \\
\qquad \land\ \text{UNCHANGED}\ \langle configuration, history \rangle \\
\lor\ \land\ transactions[i].change.commit = InProgress \\
\qquad \land\ \lor\ \land\ configuration.committed.change \neq i \\
\qquad\qquad \land\ \lor\ \land\ configuration' = [configuration\ \text{EXCEPT}\ !.committed.index\ = i, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad !.committed.change\ = i, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad !.committed.revision\ = i, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad !.committed.ordinal\ = configuration.committed. \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad !.committed.values\ = transactions[i].change.va \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad configuration.committ \\
\qquad\qquad\qquad \land\ history' = Append(history, [\\
\qquad\qquad\qquad\qquad\qquad\qquad phase\ \mapsto Change, \\
\qquad\qquad\qquad\qquad\qquad\qquad event\ \mapsto Commit, \\
\qquad\qquad\qquad\qquad\qquad\qquad index\ \mapsto i, \\
\qquad\qquad\qquad\qquad\qquad\qquad status\ \mapsto Complete]) \\
\qquad\qquad\qquad \land\ \lor\ transactions' = [transactions\ \text{EXCEPT}\ ![i].change.commit\ = Complete, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].change.ordinal\ = configuration'.committed. \\
\qquad\qquad\qquad\qquad \lor\ \text{UNCHANGED}\ \langle transactions \rangle \\
\qquad\qquad\qquad \lor\ \land\ transactions' = [transactions\ \text{EXCEPT}\ ![i].change.commit = Failed, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![i].change.apply\ = Canceled] \\
\qquad\qquad\qquad\qquad \land\ history' = Append(history, [\\
\qquad\qquad\qquad\qquad\qquad\qquad phase\ \mapsto Change, \\
\qquad\qquad\qquad\qquad\qquad\qquad event\ \mapsto Commit, \\
\qquad\qquad\qquad\qquad\qquad\qquad index\ \mapsto i,
\end{array}
$$

```
                                    status  ↦ Failed])
                        ∧ ∨ configuration′ = [configuration EXCEPT !.committed.index  = i,
                                                                   !.committed.change = i]
                          ∨ UNCHANGED ⟨configuration⟩
                    ∨ ∧ configuration.committed.change = i
                      ∧ transactions′ = [transactions EXCEPT ![i].change.commit  = Complete,
                                                             ![i].change.ordinal = configuration.committed.ordinal]
                      ∧ UNCHANGED ⟨configuration, history⟩
                ∨ ∧ transactions[i].change.commit = Failed
                  ∧ configuration.committed.change < i
                  ∧ configuration′ = [configuration EXCEPT !.committed.index  = i,
                                                           !.committed.change = i]
                  ∧ UNCHANGED ⟨transactions, history⟩
            ∧ UNCHANGED ⟨mastership, conns, target⟩

ApplyChange(n, i) ≜
    ∧ transactions[i].change.commit = Complete
    ∧ ∨ ∧ transactions[i].change.apply = Pending
        ∧ ∨ ∧ configuration.applied.ordinal = transactions[i].change.ordinal − 1
            ∧ configuration.applied.target ≠ i
            ∧ configuration.applied.index ∈ DOMAIN transactions ⇒
                ∨ ∧ configuration.applied.target = configuration.applied.index
                  ∧ transactions[configuration.applied.index].change.apply ∈ Done
                ∨ ∧ configuration.applied.target < configuration.applied.index
                  ∧ transactions[configuration.applied.index].rollback.apply ∈ Done
            ∧ ∨ ∧ configuration.applied.revision = transactions[i].rollback.index
                ∧ configuration′ = [configuration EXCEPT !.applied.target = i]
                ∧ history′ = Append(history, [
                                phase  ↦ Change,
                                event  ↦ Apply,
                                index  ↦ i,
                                status ↦ InProgress])
                ∧ ∨ transactions′ = [transactions EXCEPT ![i].change.apply = InProgress]
                  ∨ UNCHANGED ⟨transactions⟩
              ∨ ∧ configuration.applied.revision < transactions[i].rollback.index
                ∧ transactions′ = [transactions EXCEPT ![i].change.apply = Aborted]
                ∧ history′ = Append(history, [
                                phase  ↦ Change,
                                event  ↦ Apply,
                                index  ↦ i,
                                status ↦ Aborted])
                ∧ ∨ configuration′ = [configuration EXCEPT !.applied.target  = i,
                                                           !.applied.index  = i,
                                                           !.applied.ordinal = transactions[i].change.ordi
                  ∨ UNCHANGED ⟨configuration⟩
```

$\lor \land configuration.applied.target = i$
$\land transactions' = [transactions \text{ EXCEPT } ![i].change.apply = InProgress]$
$\land \text{UNCHANGED } \langle configuration, history \rangle$
$\land \text{UNCHANGED } \langle target \rangle$
$\lor \land transactions[i].change.apply = InProgress$
If the change has not yet been applied, attempt to apply it.
$\land \lor \land configuration.applied.ordinal \neq transactions[i].change.ordinal$
$\land configuration.state = Complete$
$\land configuration.term = mastership.term$
$\land conns[n].id = mastership.conn$
$\land conns[n].connected$
$\land target.running$
$\land \lor \land target' = [target \text{ EXCEPT } !.values = transactions[i].change.values \mathbin{@@} target.values]$
$\land configuration' = [configuration \text{ EXCEPT } !.applied.index = i,$
$!.applied.ordinal = transactions[i].change.ordina$
$!.applied.revision = i,$
$!.applied.values = transactions[i].change.values$
$configuration.applied.valu$
$\land history' = Append(history, [$
$phase \mapsto Change,$
$event \mapsto Apply,$
$index \mapsto i,$
$status \mapsto Complete])$
$\land \lor transactions' = [transactions \text{ EXCEPT } ![i].change.apply = Complete]$
$\lor \text{UNCHANGED } \langle transactions \rangle$
$\lor \land transactions' = [transactions \text{ EXCEPT } ![i].change.apply = Failed]$
$\land history' = Append(history, [$
$phase \mapsto Change,$
$event \mapsto Apply,$
$index \mapsto i,$
$status \mapsto Failed])$
$\land \lor configuration' = [configuration \text{ EXCEPT } !.applied.index = i,$
$!.applied.ordinal = transactions[i].change.ordi$
$\lor \text{UNCHANGED } \langle configuration \rangle$
$\land \text{UNCHANGED } \langle target \rangle$
If the change has been applied, update the transaction status.
$\lor \land configuration.applied.ordinal = transactions[i].change.ordinal$
$\land transactions' = [transactions \text{ EXCEPT } ![i].change.apply = Complete]$
$\land \text{UNCHANGED } \langle configuration, target, history \rangle$
$\lor \land transactions[i].change.apply \in \{Aborted, Failed\}$
$\land configuration.applied.ordinal < transactions[i].change.ordinal$
$\land configuration' = [configuration \text{ EXCEPT } !.applied.target = i,$
$!.applied.index = i,$
$!.applied.ordinal = transactions[i].change.ordinal]$
$\land \text{UNCHANGED } \langle transactions, target, history \rangle$

```
    ∧ UNCHANGED ⟨mastership, conns⟩

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

CommitRollback(n, i) ≜
    ∧ ∨ ∧ transactions[i].rollback.commit = Pending
        ∧ configuration.committed.revision = i
        ∧ ∨ ∧ configuration.committed.target = i
            ∧ configuration.committed.index = configuration.committed.target
            ∧ ∨ ∧ configuration.committed.index = i
                ∧ transactions[configuration.committed.index].change.commit = Complete
              ∨ ∧ configuration.committed.index > i
                ∧ transactions[configuration.committed.index].rollback.commit = Complete
            ∧ configuration' = [configuration EXCEPT !.committed.target = transactions[i].rollback.index]
            ∧ history' = Append(history, [
                                phase  ↦ Rollback,
                                event  ↦ Commit,
                                index  ↦ i,
                                status ↦ InProgress])
            ∧ ∨ transactions' = [transactions EXCEPT ![i].rollback.commit = InProgress]
              ∨ UNCHANGED ⟨transactions⟩
          ∨ ∧ configuration.committed.target = transactions[i].rollback.index
            ∧ transactions' = [transactions EXCEPT ![i].rollback.commit = InProgress]
            ∧ UNCHANGED ⟨configuration, history⟩
      ∨ ∧ transactions[i].rollback.commit = InProgress
        ∧ ∨ ∧ configuration.committed.revision = i
            ∧ configuration' = [configuration EXCEPT !.committed.index    = i,
                                                     !.committed.ordinal  = configuration.committed.ordin
                                                     !.committed.revision = transactions[i].rollback.index,
                                                     !.committed.values   = transactions[i].rollback.values
                                                                            configuration.committed.va
            ∧ history' = Append(history, [
                                phase  ↦ Rollback,
                                event  ↦ Commit,
                                index  ↦ i,
                                status ↦ Complete])
            ∧ ∨ transactions' = [transactions EXCEPT ![i].rollback.commit  = Complete,
                                                     ![i].rollback.ordinal = configuration'.committed.ordi
              ∨ UNCHANGED ⟨transactions⟩
          ∨ ∧ configuration.committed.revision = transactions[i].rollback.index
            ∧ transactions' = [transactions EXCEPT ![i].rollback.commit  = Complete,
                                                   ![i].rollback.ordinal = configuration.committed.ordinal
```

```
                ∧ UNCHANGED ⟨configuration, history⟩
        ∧ UNCHANGED ⟨mastership, conns, target⟩

ApplyRollback(n, i) ≜
    ∧ transactions[i].rollback.commit = Complete
    ∧ ∨ ∧ transactions[i].rollback.apply = Pending
        ∧ ∨ ∧ transactions[i].change.apply = Pending
            ∧ configuration.applied.ordinal = transactions[i].change.ordinal − 1
            ∧ configuration.applied.target ≠ i
            ∧ configuration.applied.index ∈ DOMAIN transactions ⇒
                ∨ ∧ configuration.applied.target = configuration.applied.index
                  ∧ transactions[configuration.applied.index].change.apply ∈ Done
                ∨ ∧ configuration.applied.target < configuration.applied.index
                  ∧ transactions[configuration.applied.index].rollback.apply ∈ Done
            ∧ transactions′ = [transactions EXCEPT ![i].change.apply = Aborted]
            ∧ history′ = Append(history, [
                                phase  ↦ Change,
                                event  ↦ Apply,
                                index  ↦ i,
                                status ↦ Aborted])
            ∧ ∨ configuration′ = [configuration EXCEPT !.applied.target  = i,
                                                       !.applied.index  = i,
                                                       !.applied.ordinal = transactions[i].change.ordinal]
              ∨ UNCHANGED ⟨configuration⟩
        ∨ ∧ transactions[i].change.apply = InProgress
            ∧ configuration.applied.ordinal ≠ transactions[i].change.ordinal
            ∧ transactions′ = [transactions EXCEPT ![i].change.apply = Failed]
            ∧ history′ = Append(history, [
                                phase  ↦ Change,
                                event  ↦ Apply,
                                index  ↦ i,
                                status ↦ Failed])
            ∧ ∨ configuration′ = [configuration EXCEPT !.applied.index  = i,
                                                       !.applied.ordinal = transactions[i].change.ordinal]
              ∨ UNCHANGED ⟨configuration⟩
        ∨ ∧ transactions[i].change.apply ∈ {Aborted, Failed}
            ∧ configuration.applied.ordinal < transactions[i].change.ordinal
            ∧ configuration′ = [configuration EXCEPT !.applied.target  = i,
                                                     !.applied.index  = i,
                                                     !.applied.ordinal = transactions[i].change.ordinal]
            ∧ UNCHANGED ⟨transactions, history⟩
        ∨ ∧ transactions[i].change.apply ∈ Done
            ∧ configuration.applied.ordinal = transactions[i].rollback.ordinal − 1
            ∧ ∨ ∧ configuration.applied.target ≠ transactions[i].rollback.index
                ∧ ∨ ∧ configuration.applied.index = i
```

$\land$ $transactions[configuration.applied.index].change.apply \in Done$
$\lor$ $\land$ $configuration.applied.index > i$
$\land$ $transactions[configuration.applied.index].rollback.apply \in Done$
$\land$ $configuration' = [configuration$ EXCEPT $!.applied.target = transactions[i].rollback.index]$
$\land$ $history' = Append(history, [$
$phase \mapsto Rollback,$
$event \mapsto Apply,$
$index \mapsto i,$
$status \mapsto InProgress])$
$\land$ $\lor$ $transactions' = [transactions$ EXCEPT $![i].rollback.apply = InProgress]$
$\lor$ UNCHANGED $\langle transactions \rangle$
$\lor$ $\land$ $configuration.applied.target = transactions[i].rollback.index$
$\land$ $transactions' = [transactions$ EXCEPT $![i].rollback.apply = InProgress]$
$\land$ UNCHANGED $\langle configuration, history \rangle$
$\land$ UNCHANGED $\langle target \rangle$
$\lor$ $\land$ $transactions[i].rollback.apply = InProgress$

If this transaction has not yet been applied, attempt to apply it.

$\land$ $\lor$ $\land$ $configuration.applied.ordinal \neq transactions[i].rollback.ordinal$
$\land$ $configuration.state = Complete$
$\land$ $configuration.term = mastership.term$
$\land$ $conns[n].id = mastership.conn$
$\land$ $conns[n].connected$
$\land$ $target.running$
$\land$ $target' = [target$ EXCEPT $!.values = transactions[i].rollback.values$ @@ $target.values]$
$\land$ $configuration' = [configuration$ EXCEPT $!.applied.index = i,$
$!.applied.ordinal = transactions[i].rollback.ordinal,$
$!.applied.revision = transactions[i].rollback.index,$
$!.applied.values = transactions[i].rollback.values$ @@
$configuration.applied.values]$
$\land$ $history' = Append(history, [$
$phase \mapsto Rollback,$
$event \mapsto Apply,$
$index \mapsto i,$
$status \mapsto Complete])$
$\land$ $\lor$ $transactions' = [transactions$ EXCEPT $![i].rollback.apply = Complete]$
$\lor$ UNCHANGED $\langle transactions \rangle$

If the change has been applied, update the transaction status.

$\lor$ $\land$ $configuration.applied.ordinal = transactions[i].rollback.ordinal$
$\land$ $configuration.applied.revision = transactions[i].rollback.index$
$\land$ $transactions' = [transactions$ EXCEPT $![i].rollback.apply = Complete]$
$\land$ UNCHANGED $\langle configuration, target, history \rangle$
$\land$ UNCHANGED $\langle mastership, conns \rangle$

$ReconcileRollback(n, i) \triangleq$
$\land$ $transactions[i].phase = Rollback$

$\land \lor CommitRollback(n,\ i)$
$\quad \lor ApplyRollback(n,\ i)$

$ReconcileTransaction(n,\ i) \triangleq$
$\land\ i \in \text{DOMAIN } transactions$
$\land\ mastership.master = n$
$\land \lor ReconcileChange(n,\ i)$
$\quad \lor ReconcileRollback(n,\ i)$