
MODULE *Transaction*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

An empty constant

CONSTANT *Nil*

Transaction phase constants

CONSTANTS

Change,

Rollback

Proposal phase constants

CONSTANTS

Commit,

Apply

Status constants

CONSTANTS

Pending,

Complete,

Canceled,

Aborted,

Failed

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

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

The set of all nodes

CONSTANT *Node*

$Empty \triangleq [p \in \{\} \mapsto Nil]$

Variables defined by other modules.

VARIABLES

configuration,

mastership,

conn,

target

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

VARIABLE *transaction*

A proposal log.

VARIABLE *proposal*

A sequence of configuration changes used for model checking.

VARIABLE *history*

$TransactionOK \triangleq$

$$\begin{aligned} & \forall i \in \text{DOMAIN } transaction : \\ & \quad \wedge transaction[i].phase \in \{Change, Rollback\} \\ & \quad \wedge transaction[i].change.proposal \in Nat \\ & \quad \wedge transaction[i].change.revision \in Nat \\ & \quad \wedge \forall p \in \text{DOMAIN } transaction[i].change.values : \\ & \quad \quad transaction[i].change.values[p] \neq Nil \Rightarrow \\ & \quad \quad transaction[i].change.values[p] \in \text{STRING} \\ & \quad \wedge transaction[i].rollback.proposal \in Nat \\ & \quad \wedge transaction[i].rollback.revision \in Nat \\ & \quad \wedge \forall p \in \text{DOMAIN } transaction[i].rollback.values : \\ & \quad \quad transaction[i].rollback.values[p] \neq Nil \Rightarrow \\ & \quad \quad transaction[i].rollback.values[p] \in \text{STRING} \end{aligned}$$

$ProposalOK \triangleq$

$$\begin{aligned} & \forall i \in \text{DOMAIN } proposal : \\ & \quad \wedge proposal[i].transaction \in Nat \\ & \quad \wedge proposal[i].commit \in Status \\ & \quad \wedge proposal[i].apply \in Status \end{aligned}$$

$TypeOK \triangleq TransactionOK \wedge ProposalOK$

$LOCAL \ State \triangleq [$

$$\begin{aligned} & \quad transactions \mapsto [i \in \text{DOMAIN } transaction \mapsto transaction[i] @@ [index \mapsto i]], \\ & \quad proposals \mapsto [i \in \text{DOMAIN } proposal \mapsto proposal[i] @@ [index \mapsto i]], \\ & \quad configuration \mapsto configuration] \end{aligned}$$

$LOCAL \ Transitions \triangleq$

LET

$$\begin{aligned} & \quad transactions \triangleq \{i \in \text{DOMAIN } transaction' : \\ & \quad \quad i \in \text{DOMAIN } transaction \Rightarrow transaction'[i] \neq transaction[i]\} \\ & \quad proposals \triangleq \{i \in \text{DOMAIN } proposal' : \\ & \quad \quad i \in \text{DOMAIN } proposal \Rightarrow proposal'[i] \neq proposal[i]\} \end{aligned}$$

IN

$$\begin{aligned} & [transactions \mapsto [i \in transactions \mapsto transaction'[i] @@ [index \mapsto i]], \\ & \quad proposals \mapsto [i \in proposals \mapsto proposal'[i] @@ [index \mapsto i]]] \end{aligned}$$

$Test \triangleq \text{INSTANCE } Test \text{ WITH}$
 $File \leftarrow \text{"Transaction.log"}$

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

Add a set of changes 'c' to the transaction log
 $AppendChange(p, v) \triangleq$
 $\wedge transaction' = Append(transaction, [$
 $\quad phase \mapsto Change,$
 $\quad change \mapsto [$
 $\quad \quad proposal \mapsto 0,$
 $\quad \quad revision \mapsto Len(transaction) + 1,$
 $\quad \quad values \mapsto (p :> v),$
 $\quad rollback \mapsto [$
 $\quad \quad proposal \mapsto 0,$
 $\quad \quad revision \mapsto 0,$
 $\quad \quad values \mapsto Empty])]$
 $\wedge \text{UNCHANGED } \langle proposal, configuration, mastership, conn, target, history \rangle$

Add a rollback of transaction 't' to the transaction log
 $RollbackChange(i) \triangleq$
 $\wedge i \in \text{DOMAIN } transaction$
 $\wedge transaction[i].phase = Change$
 $\wedge transaction[i].change.proposal \neq 0$
 $\wedge proposal[transaction[i].change.proposal].commit \neq Pending$
 $\wedge transaction' = [transaction \text{ EXCEPT } ![i].phase = Rollback]$
 $\wedge \text{UNCHANGED } \langle proposal, configuration, mastership, conn, target, history \rangle$

$ReconcileChange(n, i) \triangleq$
 The change proposal has not yet been created.
 $\wedge \vee \wedge transaction[i].change.proposal \notin \text{DOMAIN } proposal$
 The prior transaction must have created a change proposal.
 $\wedge i - 1 \in \text{DOMAIN } transaction \Rightarrow transaction[i - 1].change.proposal \in \text{DOMAIN } proposal$
 $\wedge proposal' = Append(proposal, [transaction \mapsto i, commit \mapsto Pending, apply \mapsto Pending])$
 $\wedge transaction' = [transaction \text{ EXCEPT } ![i].change.proposal = Len(proposal')]$
 $\wedge \text{UNCHANGED } \langle configuration, target, history \rangle$
 The change proposal has been created.
 $\vee \wedge transaction[i].change.proposal \in \text{DOMAIN } proposal$
 The change is pending commit. Validate and commit the change once the prior change has been committed.
 $\wedge \vee \wedge proposal[transaction[i].change.proposal].commit = Pending$
 The prior proposal has been committed.

$\wedge \text{transaction}[i].\text{change.proposal} - 1 \in \text{DOMAIN } \text{proposal} \Rightarrow$
 $\text{proposal}[\text{transaction}[i].\text{change.proposal} - 1].\text{commit} \in \text{Done}$
 The prior change has been committed.
 $\wedge \text{configuration.committed.index} = i - 1$
 Valid change is committed to the configuration.
 $\wedge \vee \wedge \text{transaction}' = [\text{transaction EXCEPT } ![i].\text{rollback.revision} = \text{configuration.committed.revision}$
 $\quad \quad \quad ![i].\text{rollback.values} = [$
 $\quad \quad \quad p \in \text{DOMAIN } \text{transaction}[i].\text{change.values} \mapsto$
 $\quad \quad \quad \text{IF } p \in \text{DOMAIN } \text{configuration.committed.values}$
 $\quad \quad \quad \text{configuration.committed.values}[p]$
 $\quad \quad \quad \text{ELSE}$
 $\quad \quad \quad \text{Nil}]$
 $\wedge \text{configuration}' = [\text{configuration EXCEPT } !.\text{committed.index} = i,$
 $\quad \quad \quad !.\text{committed.revision} = i,$
 $\quad \quad \quad !.\text{committed.values} = \text{transaction}[i].\text{change.values}$
 $\quad \quad \quad \text{configuration.committed.values}]$
 $\wedge \text{proposal}' = [\text{proposal EXCEPT } ![\text{transaction}[i].\text{change.proposal}].\text{commit} = \text{Complete}]$
 $\wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Change}, \text{phase} \mapsto \text{Commit}, \text{index} \mapsto i])$
 The change is invalid. Mark the proposal *Failed*.
 $\vee \wedge \text{configuration}' = [\text{configuration EXCEPT } !.\text{committed.index} = i]$
 $\wedge \text{proposal}' = [\text{proposal EXCEPT } ![\text{transaction}[i].\text{change.proposal}].\text{commit} = \text{Failed}]$
 $\wedge \text{UNCHANGED } \langle \text{transaction}, \text{history} \rangle$
 $\wedge \text{UNCHANGED } \langle \text{target} \rangle$
 The change was committed and apply is pending.
 $\vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{apply} = \text{Pending}$
 $\wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{commit} \in \text{Done}$
 The prior proposal has been applied.
 $\wedge \text{transaction}[i].\text{change.proposal} - 1 \in \text{DOMAIN } \text{proposal} \Rightarrow$
 $\text{proposal}[\text{transaction}[i].\text{change.proposal} - 1].\text{apply} \in \text{Done}$
 If the prior change completed apply, or if apply failed the prior change has been
 rolled back, apply this change.
 $\wedge \vee \wedge \text{configuration.applied.index} = i - 1$
 The change was committed successfully. Apply the change.
 $\wedge \vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{commit} = \text{Complete}$
 $\wedge \text{configuration.state} = \text{Complete}$
 $\wedge \text{configuration.term} = \text{mastership.term}$
 $\wedge \text{conn}[n].\text{id} = \text{mastership.conn}$
 $\wedge \text{conn}[n].\text{connected}$
 $\wedge \text{target.running}$
 The change is successfully applied to the target.
 $\wedge \vee \wedge \text{target}' = [\text{target EXCEPT } !.\text{values} = \text{transaction}[i].\text{change.values} @@ \text{target.values}]$
 $\wedge \text{configuration}' = [\text{configuration EXCEPT } !.\text{applied.index} = i,$
 $\quad \quad \quad !.\text{applied.revision} = i,$
 $\quad \quad \quad !.\text{applied.values} = \text{transaction}[i].\text{change.values}]$
 $\quad \quad \quad \text{configuration.applied.values}]$

$\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![transaction[i].change.proposal].apply = \text{Complete}]$
 $\wedge \text{history}' = \text{Append}(\text{history}, [type \mapsto \text{Change}, phase \mapsto \text{Apply}, index \mapsto i])$
 The change fails being applied to the target.
 The configuration's applied index is not incremented here to block applying
 subsequent changes until the failed change is rolled back.
 $\vee \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![transaction[i].change.proposal].apply = \text{Failed}]$
 $\wedge \text{UNCHANGED } \langle configuration, target, history \rangle$
 The change failed validation. Increment the applied index and cancel the change.
 $\vee \wedge \text{proposal}[transaction[i].change.proposal].commit = \text{Failed}$
 $\wedge \text{configuration}' = [configuration \text{ EXCEPT } !.applied.index = i]$
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![transaction[i].change.proposal].apply = \text{Canceled}]$
 $\wedge \text{UNCHANGED } \langle target, history \rangle$
 If the prior change failed apply or was aborted due to an earlier apply failure
 and the change has not been rolled back, abort this change.
 $\vee \wedge i - 1 \in \text{DOMAIN } transaction$
 $\wedge \text{proposal}[transaction[i - 1].change.proposal].apply \in \{\text{Aborted}, \text{Failed}\}$
 $\wedge \text{transaction}[i - 1].rollback.proposal \in \text{DOMAIN } proposal \Rightarrow$
 $\text{proposal}[transaction[i - 1].rollback.proposal].apply \neq \text{Complete}$
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![transaction[i].change.proposal].apply = \text{Aborted}]$
 $\wedge \text{UNCHANGED } \langle configuration, target, history \rangle$
 $\wedge \text{UNCHANGED } \langle transaction \rangle$

ReconcileRollback(n, i) \triangleq

$\wedge \text{transaction}[i].phase = \text{Rollback}$
 $\wedge \text{transaction}[i].change.proposal \in \text{DOMAIN } proposal$
 The rollback proposal has not yet been created.
 $\wedge \vee \wedge \text{transaction}[i].rollback.proposal \notin \text{DOMAIN } proposal$
 The subsequent transaction, if present, is being rolled back.
 $\wedge i + 1 \in \text{DOMAIN } transaction \Rightarrow$
 $\wedge \text{transaction}[i + 1].rollback.proposal \in \text{DOMAIN } proposal$
 $\wedge \text{Len}(proposal) = \text{transaction}[i + 1].rollback.proposal$
 $\wedge \text{proposal}' = \text{Append}(proposal, [transaction \mapsto i, commit \mapsto \text{Pending}, apply \mapsto \text{Pending}])$
 $\wedge \text{transaction}' = [\text{transaction} \text{ EXCEPT } ![i].rollback.proposal = \text{Len}(proposal')]$
 $\wedge \text{UNCHANGED } \langle configuration, target, history \rangle$
 The rollback proposal has been created.
 $\vee \wedge \text{transaction}[i].rollback.proposal \in \text{DOMAIN } proposal$
 The rollback commit is pending.
 $\wedge \vee \wedge \text{proposal}[transaction[i].rollback.proposal].commit = \text{Pending}$
 The prior proposal has been committed.
 $\wedge \text{transaction}[i].rollback.proposal - 1 \in \text{DOMAIN } proposal \Rightarrow$
 $\text{proposal}[transaction[i].rollback.proposal - 1].commit \in \text{Done}$
 The prior change has been committed.
 $\wedge i - 1 \in \text{DOMAIN } transaction \Rightarrow$
 $\text{proposal}[transaction[i - 1].change.proposal].commit \in \text{Done}$
 If the change proposal completed, commit the rollback proposal.

$\wedge \vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{commit} = \text{Complete}$
 $\wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{committed.revision} = \text{transaction}[i].\text{rollback.revision}$
 $\quad \quad \quad !.\text{committed.values} = \text{transaction}[i].\text{rollback.values}$
 $\quad \quad \quad \text{configuration.committed.values}]$
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![\text{transaction}[i].\text{rollback.proposal}].\text{commit} = \text{Complete}]$
 $\wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Rollback}, \text{phase} \mapsto \text{Commit}, \text{index} \mapsto i])$
 If the change proposal failed, complete the rollback commit.
 $\vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{commit} = \text{Failed}$
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![\text{transaction}[i].\text{rollback.proposal}].\text{commit} = \text{Complete}]$
 $\wedge \text{UNCHANGED } \langle \text{configuration}, \text{history} \rangle$
 $\wedge \text{UNCHANGED } \langle \text{target} \rangle$
 The rollback apply is pending.
 $\vee \wedge \text{proposal}[\text{transaction}[i].\text{rollback.proposal}].\text{apply} = \text{Pending}$
 The prior proposal has been applied.
 $\wedge \text{transaction}[i].\text{rollback.proposal} - 1 \in \text{DOMAIN } \text{proposal} \Rightarrow$
 $\quad \text{proposal}[\text{transaction}[i].\text{rollback.proposal} - 1].\text{apply} \in \text{Done}$
 The prior change has been applied.
 $\wedge i - 1 \in \text{DOMAIN } \text{transaction} \Rightarrow$
 $\quad \text{proposal}[\text{transaction}[i - 1].\text{change.proposal}].\text{apply} \in \text{Done}$
 The change has been applied and the rollback has been committed.
 Apply the rollback.
 $\wedge \vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{apply} \in \text{Done}$
 $\wedge \text{proposal}[\text{transaction}[i].\text{rollback.proposal}].\text{commit} = \text{Complete}$
 The change was applied or the apply failed. Ensure the rollback
 is updated in the target.
 $\wedge \vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{apply} \in \{ \text{Complete}, \text{Failed} \}$
 $\wedge \text{configuration.state} = \text{Complete}$
 $\wedge \text{configuration.term} = \text{mastership.term}$
 $\wedge \text{conn}[n].\text{id} = \text{mastership.conn}$
 $\wedge \text{conn}[n].\text{connected}$
 $\wedge \text{target.running}$
 Rollbacks are applied until successful.
 $\wedge \text{target}' = [\text{target} \text{ EXCEPT } !.\text{values} = \text{transaction}[i].\text{rollback.values} @@ \text{target.values}]$
 $\wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{applied.target} = \text{target.id},$
 $\quad \quad \quad !.\text{applied.revision} = \text{transaction}[i].\text{rollback.revision}$
 $\quad \quad \quad !.\text{applied.values} = \text{transaction}[i].\text{rollback.values}$
 $\quad \quad \quad \text{configuration.applied.values}]$
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![\text{transaction}[i].\text{rollback.proposal}].\text{apply} = \text{Complete}]$
 $\wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Rollback}, \text{phase} \mapsto \text{Apply}, \text{index} \mapsto i])$
 If the change apply was aborted or canceled, no requests were sent to the target.
 Complete the rollback apply without changes to the target.
 $\vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{apply} \in \{ \text{Aborted}, \text{Canceled} \}$
 $\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![\text{transaction}[i].\text{rollback.proposal}].\text{apply} = \text{Complete}]$
 $\wedge \text{UNCHANGED } \langle \text{configuration}, \text{target}, \text{history} \rangle$
 The change is pending apply. cancel applying the change.

$$\begin{aligned}
& \vee \wedge \text{proposal}[\text{transaction}[i].\text{change.proposal}].\text{apply} = \text{Pending} \\
& \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![\text{transaction}[i].\text{change.proposal}].\text{apply} = \text{Canceled}] \\
& \wedge \text{UNCHANGED } \langle \text{configuration}, \text{target}, \text{history} \rangle
\end{aligned}$$

If the apply is complete and the applied index matches the previous change index, increment the applied index to unblock later changes. This ensures that changes following a sequence of aborted/failed changes are blocked until the failed/aborted changes are rolled back and unblocked once all rollbacks have been applied.

$$\begin{aligned}
& \vee \wedge \text{proposal}[\text{transaction}[i].\text{rollback.proposal}].\text{apply} = \text{Complete} \\
& \wedge \text{configuration.applied.index} = i - 1 \\
& \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{applied.index} = i] \\
& \wedge \text{UNCHANGED } \langle \text{proposal}, \text{target}, \text{history} \rangle \\
& \wedge \text{UNCHANGED } \langle \text{transaction} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{ReconcileTransaction}(n, i) \triangleq \\
& \wedge i \in \text{DOMAIN } \text{transaction} \\
& \wedge \vee \text{ReconcileChange}(n, i) \\
& \quad \vee \text{ReconcileRollback}(n, i) \\
& \wedge \text{UNCHANGED } \langle \text{mastership}, \text{conn} \rangle
\end{aligned}$$