MODULE *Northbound*

```
$\textsc{extends}\ Proposal$

$\textsc{instance}\ Naturals$

$\textsc{instance}\ FiniteSets$

$\textsc{instance}\ Sequences$

$\textsc{local instance}\ TLC$
```

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

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

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

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

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

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.

```
$ValidChanges \triangleq$
  $\text{LET } changeSets \triangleq \{s \in \text{SUBSET } ValidPaths :$
                        $\land \forall p \in \text{DOMAIN } Target.values :$
                           $\land Cardinality(\{v \in s : v.path = p\}) \leq 1\}$
  $\text{IN}$
     $\{c \in \{Paths(s) : s \in changeSets\} : \text{DOMAIN } c \neq \{\}\}$
```

Add change 'c' to the proposal log

```
$Change(c) \triangleq$
  $\land \text{ LET } index \triangleq Len(proposal) + 1$
     $\text{IN } proposal' = proposal \mathbin{@@}$
           $(index :> [type \mapsto ProposalChange,$
                        index       ↦ index,
                        change      ↦ [index  ↦ index,
                                       values ↦ c],
                        rollback    ↦ [index  ↦ 0],
                        phase       ↦ ProposalValidate,
                        state       ↦ ProposalInProgress])
   ∧   UNCHANGED ⟨configuration, mastership, node, target⟩
```

Add a rollback of proposal 'i' to the proposal log

```
Rollback(i) ≜
   ∧ LET index ≜ Len(proposal) + 1
      IN  proposal′ = proposal @@
            (index :> [type        ↦ ProposalRollback,
                       index       ↦ index,
                       change      ↦ [index   ↦ 0],
                       rollback    ↦ [index   ↦ i],
                       phase       ↦ ProposalValidate,
                       state       ↦ ProposalInProgress])
   ∧ UNCHANGED ⟨configuration, mastership, node, target⟩
```

Abort aborts proposal 'i'

```
Abort(i) ≜
   ∧ proposal[i].phase ≠ ProposalAbort
   ∧ proposal[i].state  ≠ ProposalFailed
   ∧ proposal′ = [proposal EXCEPT ![i].phase = ProposalAbort,
                                  ![i].state  = ProposalInProgress]
   ∧ UNCHANGED ⟨configuration, mastership, node, target⟩
```

Formal specification, constraints, and theorems.

```
InitNorthbound ≜ TRUE

NextNorthbound ≜
   ∨ ∃ c ∈ ValidChanges :
        Change(c)
   ∨ ∃ i ∈ DOMAIN proposal :
        Rollback(i)
   ∨ ∃ i ∈ DOMAIN proposal :
        Abort(i)
```

* Modification History
* Last modified *Fri Apr* 21 16:42:15 *PDT* 2023 by *jhalterm*
* Last modified Sun *Feb* 20 10:10:06 *PST* 2022 by *jordanhalterman*
* Created Sun *Feb* 20 03:08:25 *PST* 2022 by *jordanhalterman*