
MODULE *Mastership*

EXTENDS *Southbound*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

CONSTANT *TraceMastership*

A record of target *masterships*

VARIABLE *mastership*

LOCAL *InitState* \triangleq
 [*node* \mapsto *node*,
 masterships \mapsto *mastership*]

LOCAL *NextState* \triangleq
 [*node* \mapsto *node'*,
 masterships \mapsto *mastership'*]

LOCAL *Trace* \triangleq INSTANCE *Trace* WITH
 Module \leftarrow "Mastership",
 InitState \leftarrow *InitState*,
 NextState \leftarrow *NextState*,
 Enabled \leftarrow *TraceMastership*

This section models *mastership* reconciliation.

ReconcileMastership(*n*) \triangleq
 $\wedge \vee \wedge$ *node*[*n*].*connected*
 \wedge *mastership*.*master* = *Nil*
 \wedge *mastership'* = [*master* \mapsto *n*, *term* \mapsto *mastership.term* + 1]
 $\vee \wedge \neg$ *node*[*n*].*connected*
 \wedge *mastership*.*master* = *n*
 \wedge *mastership'* = [*mastership* EXCEPT !.*master* = *Nil*]
 \wedge UNCHANGED \langle *node*, *target* \rangle

Formal specification, constraints, and theorems.

InitMastership \triangleq
 \wedge *mastership* = [*master* \mapsto *Nil*, *term* \mapsto 0]

$$NextMastership \triangleq$$

$$\forall \exists n \in Node :$$

$$Trace! Step(ReconcileMastership(n), [node \mapsto n])$$

\ * Modification History
\ * Last modified Sun Feb 20 09:09:52 PST 2022 by jordanhalterman
\ * Created Sun Feb 20 03:13:26 PST 2022 by jordanhalterman