```
1  ---------- MODULE stateconsistency ----------
2  EXTENDS Naturals, TLC, Sequences, FiniteSets, Integers, StatesConsistencyInspect, SnapFileSync, Broadcast
3   Sequence to Set
4  RECURSIVE Seq2Set(_)
5  Seq2Set(S) ≜
6      IF S = ⟨⟩ THEN {}
7       ELSE
8          LET i ≜ Head(S)
9          IN   {i} ∪ Seq2Set(Tail(S))

12 IsInjective(f) ≜ ∀ a, b ∈ DOMAIN f : f[a] = f[b] ⇒ a = b
13 SetToSeq(S) ≜ CHOOSE f ∈ [1 .. Cardinality(S) → S] : IsInjective(f)

15 decideStateIsLeading(S) ≜ ∃ x ∈ S :
16                                 x.State = "LEADING"


20 selectStateIsLeading2(S) ≜ CHOOSE x ∈ S :
21                                     x.State = "LEADING"
22 selectStateIsNotLeading(S) ≜ {x ∈ S :
23                                     x.State ≠ "LEADING"}

26 decideStateIsfOLLOWING(S, myid) ≜ ∀ x ∈ S :
27                                     ∧ (x.State = "FOLLOWING") ∨ (x.State = "LOOKING")
28                                     ∧ x.myId ≠ myid


31 VARIABLES stateSeq, offset, statecollection, myId
32 vars ≜ ⟨stateSeq, statecollection, offset, myId⟩
33 Trace ≜ StateConsistencyParser("D:\\00001code\\runtimemodel\\zookeeper_environment\\stateconsistency.lo

35 Init  ≜  ∧ offset = 1
36          ∧ stateSeq = ⟨⟩
37          ∧ statecollection = {}
38          ∧ myId = 0
```

$$term \triangleq \land offset > Len(Trace)$$
$$\land \text{UNCHANGED } vars$$

Input data:⟨[$myId \mapsto 1$, $State \mapsto$ "FOLLOWING"], [$myId \mapsto 2$, $State \mapsto$ "LEADING"]⟩

$$selectStateIsLeading(S) \triangleq \{x \in S : x.State = \text{"LEADING"}\}$$

$$Rule2 \triangleq \land offset \leq Len(StateConsistencyParser(\text{"./stateconsistency.log"}))$$
$$\land offset' = offset + 1$$
$$\land stateSeq' = StateConsistencyParser(\text{"./stateconsistency.log"})[offset]$$
$$\land statecollection' = Seq2Set(stateSeq')$$
$$\land Assert(Len(SetToSeq(selectStateIsLeading(statecollection'))) = 1, \text{"Leader is one node"})$$

$$stateconsistency \triangleq \land offset \leq Len(Trace)$$
$$\land offset' = offset + 1$$
$$\land stateSeq' = Trace[offset]$$
$$\land statecollection' = Seq2Set(stateSeq')$$
$$\land Assert(decideStateIsLeading(statecollection') = \text{TRUE}, \text{"Existed state is Lead}$$
$$\land Assert(Len(SetToSeq(selectStateIsLeading(statecollection'))) = 1, \text{"Leader is}$$
$$\land myId' = selectStateIsLeading2(statecollection').myId$$
$$\land Assert(decideStateIsfOLLOWING(selectStateIsNotLeading(statecollection'),$$

$$Next \triangleq \lor stateconsistency$$
$$\lor term$$

$$Spec \triangleq Init \land \Box[Next]_{vars}$$

* Modification History
* Last modified *Wed Mar* 16 18:21:20 *CST* 2022 by 10222803
* Created *Fri Feb* 25 14:49:26 *CST* 2022 by 10222803