```
MODULE Counter

TLA+ module for Op-based Counter. See its implementation in paper Burckhardt@POPL'2014.

EXTENDS Naturals, Sequences

CONSTANTS
    Replica  the set of replicas

VARIABLES
    counter,   counter[r]: the current value of the counter at replica r ∈ Replica
    acc,       acc[r]: the number of increments performed since the last broadcast at replica r ∈ Replica
    incoming   incoming[r]: incoming messages at replica r ∈ Replica

TypeOK ≜  ∧ counter ∈ [Replica → Nat]
          ∧ acc ∈ [Replica → Nat]
          ∧ incoming ∈ [Replica → Seq(Nat)]

Init ≜  ∧ counter = [r ∈ Replica ↦ 0]
        ∧ acc = [r  ∈ Replica ↦ 0]
        ∧ incoming = [r ∈ Replica ↦ ⟨⟩]

Inc() at replica r ∈ Replica

Inc(r) ≜  ∧ TRUE    no pre-cond
          ∧ counter′ = [counter EXCEPT ![r] = @ + 1]    current counter + 1
          ∧ acc′ = [acc EXCEPT ![r] = @ + 1]    # of increments + 1
          ∧ UNCHANGED ⟨incoming⟩

 broadcast a message to all replicas except the sender s
Broadcast(s, m) ≜ [r ∈ Replica ↦
                      IF s = r
                       THEN incoming[s]
                       ELSE Append(incoming[r], m)]

Send(r) ≜  ∧ acc[r] ≠ 0
           ∧ acc′ = [acc EXCEPT ![r] = 0]   reset acc[r]
           ∧ Broadcast(r, acc[r])
           ∧ UNCHANGED ⟨counter⟩

Receive(r) ≜  ∧ incoming[r] ≠ ⟨⟩
              ∧ LET m = Head(incoming[r])
                IN  counter′ = [counter EXCEPT ![r] = @ + m]
              ∧ incoming′ = [incoming EXCEPT ![r] = Tail(@)]
              ∧ UNCHANGED ⟨acc⟩

Next ≜ ∃ r ∈ Replica : Inc(r) ∨ Send(r) ∨ Receive(r)

vars = ⟨counter, acc, incoming⟩
```

55 $Spec \triangleq Init \wedge \Box[Next]_{vars}$

56

* Modification History
* Last modified Sun *Jun* 03 21:55:15 *CST* 2018 by *hengxin*
* Created Sun *Jun* 03 20:08:57 *CST* 2018 by *hengxin*

* *oid*, * *oid*[*r*]: the (local) *id* of the increment operation issued by the replica $r \in Replica$
* *oids*, * *oids*[*r*]: the set of ids of the increment operations performed by the replica $r \in Replica$

* $Oid \triangleq$ [*r*: *Replica*, *id*: *Nat*]
* $Msg \triangleq$ [a: *Nat*, *r*: *Replica*, *id*: *Nat*] * message to send