MODULE *Integrity*

EXTENDS *Naturals*, *FiniteSets*, *Commons*

CONSTANT *NPROCESSES*
CONSTANT *NMESSAGES*
CONSTANT *CONFLICTR*(_, _)

Since this algorithm is for failure-free environments, the set of all processes is the same as the correct ones.

LOCAL $Processes \triangleq \{i : i \in 1 \,..\, NPROCESSES\}$
LOCAL $ChooseProcess \triangleq$ CHOOSE $x \in Processes$: TRUE

This property verifies that we only deliver sent messages. To assert this, we create *NMESSAGES*+ 1 and do not include the additional one in the algorithm execution, then check that the delivered ones are only the sent ones.

LOCAL $AcceptableMessageIds \triangleq \{id : id \in 1 \,..\, NMESSAGES\}$
LOCAL $Create(id) \triangleq [id \mapsto id,\ d \mapsto Processes,\ o \mapsto ChooseProcess]$
LOCAL $AllMessages \triangleq \{Create(id) : id \in 1 \,..\, (NMESSAGES + 1)\}$
LOCAL $SentMessage \triangleq \{m \in AllMessages : m.id \in AcceptableMessageIds\}$

VARIABLES K, $Pending$, $Delivering$, $Delivered$,
$PreviousMsgs$, $Votes$, $QuasiReliableChannel$

Initialize the instance for the Generic Multicast 0. The *INITIAL_MESSAGES* is a set with *NMESSAGES*, unordered, a tuple with the starting state *S*0 and the message.

$Algorithm \triangleq$ INSTANCE $GenericMulticast0$ WITH
$INITIAL\_MESSAGES \leftarrow \{\langle \text{“S0”}, m\rangle : m \in SentMessage\}$

Weak fairness is necessary.

$Spec \triangleq Algorithm!SpecFair$

LOCAL $DeliveredOnlyOnce(p, m) \triangleq$
$Cardinality(Algorithm!FilterDeliveredMessages(p, m)) = 1$

For every message, all the correct processes in the destination deliver it only once, and a process previously sent it.

$Integrity \triangleq$
$\Diamond\Box\forall\, m \in AllMessages :$
$\forall\, p \in m.d :$
$(p \in Processes \land DeliveredOnlyOnce(p, m)) \equiv m \in SentMessage$