MODULE *PartialOrder*

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

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 $S0$ and the message.

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

$Spec \triangleq Algorithm!Spec$

LOCAL $BothDelivered(p, q, m, n) \triangleq$
$\land\ Algorithm!WasDelivered(p, m) \land Algorithm!WasDelivered(p, n)$
$\land\ Algorithm!WasDelivered(q, m) \land Algorithm!WasDelivered(q, n)$

LOCAL $LHS(p, q, m, n) \triangleq$
$\{p, q\} \subseteq (m.d \cap n.d) \land BothDelivered(p, q, m, n) \land CONFLICTR(m, n)$

LOCAL $RHS(p, q, m, n) \triangleq$
$(Algorithm!DeliveredInstant(p, m) < Algorithm!DeliveredInstant(p, n))$
$\equiv (Algorithm!DeliveredInstant(q, m) < Algorithm!DeliveredInstant(q, n))$

For every two messages, if they conflict, given a pair of processes, they are in the messages' destination, then both must deliver in the same order.

$PartialOrder \triangleq$
$\Box\forall\, p, q \in Processes :$
$\forall\, m, n \in AllMessages :$
$LHS(p, q, m, n) \Rightarrow RHS(p, q, m, n)$