MODULE *PartialOrder*

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

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

LOCAL $Processes \triangleq 1 \,..\, NPROCESSES$
LOCAL $Groups \triangleq 1 \,..\, NGROUPS$
LOCAL $ProcessesInGroup \triangleq [g \in Groups \mapsto Processes]$

LOCAL $AllMessages \triangleq CreateMessages(NMESSAGES, Groups, Processes)$
LOCAL $MessagesCombinations \triangleq CreatePossibleMessages(AllMessages)$

VARIABLES K, $PreviousMsgs$, $Delivered$, $Votes$, $MemoryBuffer$,
$QuasiReliableChannel$, $AtomicBroadcastBuffer$

Initialize the instance for the Generic Multicast 1. The *INITIAL_MESSAGES* is a sequence, totally ordered within a group, wherein the elements are tuples with the message, state, and timestamp.

$Algorithm \triangleq$ INSTANCE $GenericMulticast1$ WITH
$INITIAL\_MESSAGES \leftarrow [g \in Groups \mapsto$
$TotallyOrdered(MessagesCombinations[(g\%NMESSAGES) + 1])]$

$Spec \triangleq Algorithm!Spec$

LOCAL $BothDelivered(g, p1, p2, m1, m2) \triangleq$
$\land Algorithm!WasDelivered(g, p1, m1) \land Algorithm!WasDelivered(g, p1, m2)$
$\land Algorithm!WasDelivered(g, p2, m1) \land Algorithm!WasDelivered(g, p2, m2)$

LOCAL $LHS(g, p1, p2, m1, m2) \triangleq$
$\land \{p1, p2\} \subseteq (m1.d \cap m2.d)$
$\land CONFLICTR(m1, m2)$
$\land BothDelivered(g, p1, p2, m1, m2)$

LOCAL $RHS(g, p1, p2, m1, m2) \triangleq$
$(Algorithm!DeliveredInstant(g, p1, m1) <$
$Algorithm!DeliveredInstant(g, p1, m2))$
$\equiv (Algorithm!DeliveredInstant(g, p2, m1) <$
$Algorithm!DeliveredInstant(g, p2, m2))$

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.

$PartialOrder \triangleq$
$\Box \forall\, g \in Groups :$
$\forall\, p1, p2 \in ProcessesInGroup[g] :$
$\forall\, m1, m2 \in AllMessages :$
$LHS(g, p1, p2, m1, m2) \Rightarrow RHS(g, p1, p2, m1, m2)$