$$\text{MODULE } RaftHeartbeat$$

EXTENDS *Naturals*, *FiniteSets*, *Sequences*, *TLC*

Is leader *ALIVE* or *CRASHED*
VARIABLE *leaderState*

A collection of heartbeat (*AppendEntries*) messages the leader has sent.
A single message is abstracted to represent the leader's index
VARIABLE *messages*

A representation of the *commitIndex* and term, leader increases index monotonically.
VARIABLE *leaderIndex*
VARIABLE *followerIndex*
$nodeIndexes \triangleq \langle leaderIndex, followerIndex \rangle$

Indicates whether the follower timed out after not hearing from
the leader for the specified amount of time.
VARIABLE *isTimeout*

$vars \triangleq \langle leaderState, messages, nodeIndexes, isTimeout \rangle$

The leader crashes and doesn't recover
$CrashLeader \triangleq$
  $\land leaderState = \text{"ALIVE"}$
  $\land leaderState' = \text{"CRASHED"}$
  $\land$ UNCHANGED $\langle messages, nodeIndexes, isTimeout \rangle$

The leader sends the follower an *AppendEntries* message
$SendMessage \triangleq$
  $\land leaderState = \text{"ALIVE"}$
  $\land messages' = Append(messages, leaderIndex)$
  $\land$ UNCHANGED $\langle leaderState, nodeIndexes, isTimeout \rangle$

Helper function to remove a message from a sequence of messages
$RemoveMessage(i, seq) \triangleq$
 $[j \in 1 .. Len(seq) - 1 \mapsto \text{IF } j < i \text{ THEN } seq[j] \text{ ELSE } seq[j + 1]]$

The network drops a message
$DropMessage \triangleq$
  $\land Len(messages) \geq 1$
  $\land \exists i \in 1 .. Len(messages):$
   $messages' = RemoveMessage(i, messages)$
  $\land$ UNCHANGED $\langle leaderState, nodeIndexes, isTimeout \rangle$

The leader increments its index
$IncrementIndex \triangleq$
  $\land leaderState = \text{"ALIVE"}$

1

$\land\ leaderIndex' = leaderIndex + 1$
$\land\ \text{UNCHANGED}\ \langle leaderState,\ messages,\ followerIndex,\ isTimeout\rangle$

The follower receives a message from the leader.
$ReceiveMessage\ \triangleq$
$\qquad \land\ Len(messages) \geq 1$
$\qquad \land\ \exists\, i \in 1\,..\, Len(messages):$
$\qquad\qquad ((\text{LET}\ message\ \triangleq\ messages[i]$
$\qquad\qquad\ \ \text{IN}\quad followerIndex' = \text{IF}\ message > followerIndex$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{THEN}\ message$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE}\ followerIndex)$
$\qquad\qquad\ \ \land\quad messages' = RemoveMessage(i,\ messages))$
$\qquad \land\ \text{UNCHANGED}\ \langle leaderState,\ leaderIndex,\ isTimeout\rangle$

The follower times out
$Timeout\ \triangleq\ isTimeout' = \text{TRUE}$
$\qquad\qquad \land\quad \text{UNCHANGED}\ \langle leaderState,\ messages,\ nodeIndexes\rangle$

Initial state of model
$Init\ \triangleq\ \land\ leaderState = \text{``ALIVE''}$
$\qquad\quad \land\ messages = \langle\rangle$
$\qquad\quad \land\ leaderIndex = 0$
$\qquad\quad \land\ followerIndex = 0$
$\qquad\quad \land\ isTimeout = \text{FALSE}$

Next state function
$Next\ \triangleq\ \lor\ SendMessage$
$\qquad\qquad \lor\ IncrementIndex$
$\qquad\qquad \lor\ DropMessage$
$\qquad\qquad \lor\ ReceiveMessage$
$\qquad\qquad \lor\ CrashLeader$
$\qquad\qquad \lor\ Timeout$

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