```
──────────────────── 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 ≜ ⟨leaderIndex, followerIndex⟩

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

vars ≜ ⟨leaderState, messages, nodeIndexes, isTimeout⟩

  The leader crashes and doesn't recover
CrashLeader ≜
          ∧ leaderState = "ALIVE"
          ∧ leaderState′ = "CRASHED"
          ∧ UNCHANGED ⟨messages, nodeIndexes, isTimeout⟩

  The leader sends the follower an AppendEntries message
SendMessage ≜
          ∧ leaderState = "ALIVE"
          ∧ messages′ = Append(messages, leaderIndex)
          ∧ UNCHANGED ⟨leaderState, nodeIndexes, isTimeout⟩

  Helper function to remove a message from a sequence of messages
RemoveMessage(i, seq) ≜
    [j ∈ 1 . . Len(seq) − 1 ↦ IF j < i THEN seq[j] ELSE  seq[j + 1]]

  The network drops a message
DropMessage ≜
          ∧ Len(messages) ≥ 1
          ∧ ∃ i ∈ 1 . . Len(messages) :
              messages′ = RemoveMessage(i, messages)
          ∧ UNCHANGED ⟨leaderState, nodeIndexes, isTimeout⟩

  The leader increments its index
IncrementIndex ≜
          ∧ leaderState = "ALIVE"
```

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

The follower receives a message from the leader.

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

The follower times out

```
$Timeout\ \triangleq\ isTimeout' = \text{TRUE}$
          $\land\quad \text{UNCHANGED}\ \langle leaderState,\ messages,\ nodeIndexes \rangle$
```

Initial state of model

```
$Init\ \triangleq\ \land\ leaderState = \text{“ALIVE”}$
          $\land\ messages = \langle\rangle$
          $\land\ leaderIndex = 0$
          $\land\ followerIndex = 0$
          $\land\ isTimeout = \text{FALSE}$
```

Next state function

```
$Next\ \triangleq\ \lor\ SendMessage$
           $\lor\ IncrementIndex$
           $\lor\ DropMessage$
           $\lor\ ReceiveMessage$
           $\lor\ CrashLeader$
           $\lor\ Timeout$
```

```
$Spec\ \triangleq\ Init \land \Box[Next]_{vars} \land \mathrm{WF}_{vars}(Next)$
```

Invariant that helps make sure we haven't stepped out of bounds

```
$TypeOK\ \triangleq\ \land\ leaderState \in \{\text{“ALIVE”},\ \text{“CRASHED”}\}$
             $\land\ messages \in Seq(Nat)$
             $\land\ leaderIndex \in Nat$
             $\land\ followerIndex \in Nat$
             $\land\ isTimeout \in \text{BOOLEAN}$
```

Properties of the system

```
$LeaderFailureDetected\ \triangleq\ leaderState = \text{“CRASHED”} \leadsto isTimeout = \text{TRUE}$

$\text{THEOREM}\ Correctness\ \triangleq\ Spec \Rightarrow \Box LeaderFailureDetected$
```