MODULE *syncCon1*

```
EXTENDS $Integers, Sequences, FiniteSets, TLC$
CONSTANTS $N, FAILNUM$
ASSUME $N \leq 5 \land 0 \leq FAILNUM \land FAILNUM \leq 4$
$Nodes \triangleq 1 \,..\, N$
```

```
--algorithm $syncCon1$
{ variable $FailNum = FAILNUM$,
            $up = [n \in Nodes \mapsto \text{TRUE}]$ ;
            $pt = [n \in Nodes \mapsto 0]$ ;
            $t = [n \in Nodes \mapsto \text{FALSE}]$ ;
            $d = [n \in Nodes \mapsto -1]$ ;
            $mb = [n \in Nodes \mapsto \{\}]$ ;
  define {
     $SetMin(S) \triangleq \text{CHOOSE } i \in S : \forall j \in S : i \leq j$
   }
  macro $MayBeFail(\;)$ {
     if ( $FailNum > 0 \land up[self]$ ) {
          either {
                   $up[self] := \text{FALSE}$ ;
                   $FailNum := FailNum - 1$ ;
                   }
          or skip ;
      } ;
   }

  fair process ( $n \in Nodes$ )
  variable $v = 0,\ pv = 0,\ Q = \{\}$ ;
  {
  $P$: if ( $up[self]$ ) {
         $v := self$ ;
         $Q := Nodes$ ;

  $PS$:    while ( $up[self] \land Q \neq \{\}$ ) {
             with ( $p \in Q$ ) {
                Node can fail here, such that $up[self]$ will be set to False
                A process can fail anytime during the broadcast
               $MayBeFail()$ ;
                Pop process $p$ from $Q$
               $Q := Q \setminus \{p\}$ ;
               if ( $up[self]$ ) {
                    $mb[p] := mb[p] \cup \{v\}$ ;    Broadcast value of self to each process if self is up
                 }
              } ;
            } ;
          A node may fail after broadcast
          MayBeFail() ;

          Increase the number of rounds that are completed if the node is up
          if ( up[self] ) {
               pt[self] := pt[self] + 1 ;
           } ;

           To await, a process must be up and every other process should be on the same round
           Wait for others to move to next round
           If the node is down, exit.
 PR:      await (up[self] = FALSE ∨ (up[self] ∧ (∀ i ∈ Nodes : IF up[i] THEN pt[i] = pt[self] ELSE TRUE))) ;

           Terminate and compute decision if the node is up
          if ( up[self] ) {
               d[self] := SetMin(mb[self]) ;
               t[self] := TRUE ;
           }
      } ;
    }
 }
 BEGIN TRANSLATION
VARIABLES FailNum, up, pt, t, d, mb, pc

 define statement
SetMin(S) ≜ CHOOSE i ∈ S : ∀ j ∈ S : i ≤ j

VARIABLES v, pv, Q

vars ≜ ⟨FailNum, up, pt, t, d, mb, pc, v, pv, Q⟩

ProcSet ≜ (Nodes)

Init ≜  Global variables
         ∧ FailNum = FAILNUM
         ∧ up = [n  ∈ Nodes ↦ TRUE]
         ∧ pt  = [n  ∈ Nodes ↦ 0]
         ∧ t  = [n ∈ Nodes ↦ FALSE]
         ∧ d = [n ∈ Nodes ↦  − 1]
         ∧ mb = [n ∈ Nodes ↦ {}]
          Process n
         ∧ v = [self ∈ Nodes ↦ 0]
         ∧ pv = [self ∈ Nodes ↦ 0]
         ∧ Q = [self ∈ Nodes ↦ {}]
         ∧ pc = [self ∈ ProcSet ↦ "P"]

P(self) ≜  ∧ pc[self] = "P"
```

```
        $\land$ IF $up[self]$
              THEN $\land\ v' = [v$ EXCEPT $![self] = self]$
                   $\land\ Q' = [Q$ EXCEPT $![self] = Nodes]$
                   $\land\ pc' = [pc$ EXCEPT $![self] = \text{“PS”}]$
              ELSE $\land\ pc' = [pc$ EXCEPT $![self] = \text{“Done”}]$
                   $\land$ UNCHANGED $\langle v, Q \rangle$
        $\land$ UNCHANGED $\langle FailNum, up, pt, t, d, mb, pv \rangle$

$PS(self) \triangleq$ $\land\ pc[self] = \text{“PS”}$
          $\land$ IF $up[self] \land Q[self] \neq \{\}$
                THEN $\land\ \exists\, p \in Q[self]$ :
                          $\land$ IF $FailNum > 0 \land up[self]$
                                THEN $\land$ $\lor$ $\land\ up' = [up$ EXCEPT $![self] =$ FALSE$]$
                                             $\land\ FailNum' = FailNum - 1$
                                         $\lor$ $\land$ TRUE
                                             $\land$ UNCHANGED $\langle FailNum, up \rangle$
                                ELSE $\land$ TRUE
                                     $\land$ UNCHANGED $\langle FailNum, up \rangle$
                          $\land\ Q' = [Q$ EXCEPT $![self] = Q[self] \setminus \{p\}]$
                          $\land$ IF $up'[self]$
                                THEN $\land\ mb' = [mb$ EXCEPT $![p] = mb[p] \cup \{v[self]\}]$
                                ELSE $\land$ TRUE
                                     $\land\ mb' = mb$
                     $\land\ pc' = [pc$ EXCEPT $![self] = \text{“PS”}]$
                     $\land\ pt' = pt$
                ELSE $\land$ IF $FailNum > 0 \land up[self]$
                           THEN $\land$ $\lor$ $\land\ up' = [up$ EXCEPT $![self] =$ FALSE$]$
                                        $\land\ FailNum' = FailNum - 1$
                                    $\lor$ $\land$ TRUE
                                        $\land$ UNCHANGED $\langle FailNum, up \rangle$
                           ELSE $\land$ TRUE
                                $\land$ UNCHANGED $\langle FailNum, up \rangle$
                     $\land$ IF $up'[self]$
                           THEN $\land\ pt' = [pt$ EXCEPT $![self] = pt[self] + 1]$
                           ELSE $\land$ TRUE
                                $\land\ pt' = pt$
                     $\land\ pc' = [pc$ EXCEPT $![self] = \text{“PR”}]$
                     $\land$ UNCHANGED $\langle mb, Q \rangle$
          $\land$ UNCHANGED $\langle t, d, v, pv \rangle$

$PR(self) \triangleq$ $\land\ pc[self] = \text{“PR”}$
          $\land\ (up[self] =$ FALSE $\lor\ (up[self] \land (\forall\, i \in Nodes :$ IF $up[i]$ THEN $pt[i] = pt[self]$ ELSE TRUE$)))$
          $\land$ IF $up[self]$
                THEN $\land\ d' = [d$ EXCEPT $![self] = SetMin(mb[self])]$
                     $\land\ t' = [t$ EXCEPT $![self] =$ TRUE$]$
```

$$\begin{array}{lll} & \text{ELSE} & \land\ \text{TRUE} \\ & & \land\ \text{UNCHANGED}\ \langle t,\ d\rangle \\ \land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“Done”}] & & \\ \land\ \text{UNCHANGED}\ \langle FailNum,\ up,\ pt,\ mb,\ v,\ pv,\ Q\rangle & & \end{array}$$

$$n(self) \triangleq P(self) \lor PS(self) \lor PR(self)$$

$$Next \triangleq (\exists\, self \in Nodes : n(self))$$

$\lor$ Disjunct to prevent deadlock on termination

$$((\forall\, self \in ProcSet : pc[self] = \text{“Done”}) \land \text{UNCHANGED}\ vars)$$

$$\begin{aligned} Spec \triangleq\ & \land\ Init \land \Box[Next]_{vars} \\ & \land\ \forall\, self \in Nodes : \mathrm{WF}_{vars}(n(self)) \end{aligned}$$

Agreement invariant

The nodes that have terminated (up nodes) have the same decision values

$$Agreement \triangleq \forall\, i \in Nodes : \forall\, j \in Nodes : \text{IF}\ t[i] \land t[j]\ \text{THEN}\ d[i] = d[j]\ \text{ELSE}\ \ \text{TRUE}$$

Termination property

If the node is up, it has terminated (its t is TRUE) otherwise it has not terminated

$$Termination \triangleq \Diamond(\forall\, i \in Nodes : \text{IF}\ up[i]\ \text{THEN}\ t[i] = \text{TRUE ELSE}\ \ t[i] = \text{FALSE})$$

END TRANSLATION

END TRANSLATION

Members:

Name: $Sneha\ Mehta\ UBIT$ Name: snehameh Person $\neq\ -50245877$

Name: $Varun\ Jain\ UBIT$ Name: varunjai Person $\neq\ -50247176$

Explanation: Agreement Property: The nodes that have terminated (up nodes) have the same decision values

- The agreement property is satisfied when $FAILNUM = 0$

We take 3 $Nodes$ in the system and check for $FAILNUM = 1$ ie 1 node crashes out of the 3 nodes. In this scenario the final state at which the $Agreement$ Invariant is violated is:

$\land\ FailNum = 0$
$\land\ Q = \langle\{3\},\ \{\},\ \{\}\rangle$
$\land\ d = \langle -1,\ 1,\ 2\rangle$
$\land\ mb = \langle\{2,\ 3\},\ \{1,\ 2,\ 3\},\ \{2,\ 3\}\rangle$
$\land\ pc = \langle\text{“PS”},\ \text{“Done”},\ \text{“Done”}\rangle$
$\land\ pt = \langle 0,\ 1,\ 1\rangle$
$\land\ pv = \langle 0,\ 0,\ 0\rangle$
$\land\ t = \langle\text{FALSE},\ \text{TRUE},\ \text{TRUE}\rangle$
$\land\ up = \langle\text{FALSE},\ \text{TRUE},\ \text{TRUE}\rangle$
$\land\ v = \langle 1,\ 2,\ 3\rangle$

1. In this scenario, when Node 1 fails, it was able to broadcast its value to Node 2 but not to Node 3.
2. Node 2 *terminates*$(t = \text{TRUE})$ with minimum value 1, which was sent by Node 1 (its decision is 1).
3. Node 3 *terminates*$(t = \text{TRUE})$ with minimum value 2 (its decision is 2).
4. Since both have different decision values at termination, it violates the *Agreement* property.
5. Note that Node 1 fails, and we do not set its $t[1]$ to TRUE. It remains FALSE as initially set. The decision of the crashed node is not taken into consideration.

Nodes: 3 and Crash: 2

$\land\ FailNum = 1$
$\land\ Q = \langle \{3\},\ \{\},\ \{\} \rangle$
$\land\ d = \langle -1,\ 1,\ 2 \rangle$
$\land\ mb = \langle \{2,\ 3\},\ \{1,\ 2,\ 3\},\ \{2,\ 3\} \rangle$
$\land\ pc = \langle \text{“PS”},\ \text{“Done”},\ \text{“Done”} \rangle$
$\land\ pt = \langle 0,\ 1,\ 1 \rangle$
$\land\ pv = \langle 0,\ 0,\ 0 \rangle$
$\land\ t = \langle \text{FALSE},\ \text{TRUE},\ \text{TRUE} \rangle$
$\land\ up = \langle \text{FALSE},\ \text{TRUE},\ \text{TRUE} \rangle$
$\land\ v = \langle 1,\ 2,\ 3 \rangle$