MODULE *syncCon2*

zheng kai 50247576
xinbo yu 50102922
Synchronized Consensus

EXTENDS *Integers*, *Sequences*, *FiniteSets*, *TLC*
CONSTANTS N, *FAILNUM*
ASSUME $N \leq 5 \land 0 \leq FAILNUM \land FAILNUM \leq 4$
$Nodes \triangleq 1 \,..\, N$
$counter \triangleq 0$ record the *FailNum*

```
--algorithm syncCon2{
    variables FailNum = FAILNUM ;                Initialization block
              up = [n  ∈ Nodes ↦ TRUE] ;         Nodes are up
              pt = [n  ∈ Nodes ↦ 0] ;            Nodes are at round 0
              t = [n ∈ Nodes ↦ FALSE] ;          Nodes are not terminated
              d = [n ∈ Nodes ↦ −1] ;             Nodes are not decided
              mb = [n ∈ Nodes ↦ {}] ;            Nodes have mailbox as emptyset

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

    macro MaybeFail( ) {
        if ( FailNum > 0 ∧ up[self] )
        {
            either { up[self] := FALSE ; FailNum := FailNum − 1 ; counter := counter + 1 }   if fail, counter +
            or skip ;
         }
     }

    fair process ( n ∈ Nodes )
    variables v = 0, Q = {} ;
    {
        P: if ( up[self] ) {
                v := self ;
                Q := Nodes ;
           PS: while ( pt[self] ≤ counter ) {
                    if ( pt[self] = 0 ) {
                        Q := Nodes ;
                        L1: while ( up[self] ∧ Q ≠ {} ) {
                            with ( p ∈ Q ) {
                                MaybeFail() ;     the node can crash when sending message
                                if ( ¬up[self] ) {          if the node crash, it's terminated
                                    t[self] := TRUE ;
                                    goto PR ;
                                 }
```

```
                        else {
                            mb[p] := mb[p] ∪ {v} ;      put value v into the node's mailbox
                            Q := Q \ {p} ;              delete p out of the set Q
                        }
                    } ;
                } ;
            }
            else {
                await (∀ i ∈ Nodes : pt[i] ≥ 1 ∨ ¬up[i]) ;      for all nodes, if not at at round 0, or up[i] = Fa
                Q := Nodes ;
                L2: while ( up[self] ∧ Q ≠ {} ) {
                    with ( p ∈ Q ) {
                        MaybeFail() ;
                        if ( ¬up[self] ) {
                            t[self] := TRUE ;
                            goto PR ;
                        }
                        else {
                            mb[p] := mb[p] ∪ mb[self] ;
                            Q := Q \ {p} ;
                        }
                    } ;
                } ;
            } ;
            L3:   pt[self] := pt[self] + 1 ;
        } ;
    PR: await (∀ n ∈ Nodes : (pt[n] = counter + 1 ∨ up[n] = FALSE)) ;      await for all nodes at rounds coun
        if ( up[self] ) d[self] := SetMin(mb[self]) ;
        t[self] := TRUE ;
    }
  }
}
```

BEGIN TRANSLATION

VARIABLES $FailNum$, up, pt, t, d, mb, pc

define statement

$SetMin(S) \triangleq$ CHOOSE $i \in S : \forall j \in S : i \leq j$

VARIABLES v, Q

$vars \triangleq \langle FailNum, up, pt, t, d, mb, pc, v, Q \rangle$

$ProcSet \triangleq (Nodes)$

$Init \triangleq$ Global variables
$\wedge\ FailNum = FAILNUM$

$\land\ up = [n \in Nodes \mapsto \text{TRUE}]$
$\land\ pt = [n \in Nodes \mapsto 0]$
$\land\ t = [n \in Nodes \mapsto \text{FALSE}]$
$\land\ d = [n \in Nodes \mapsto -1]$
$\land\ mb = [n \in Nodes \mapsto \{\}]$
Process n
$\land\ v = [self \in Nodes \mapsto 0]$
$\land\ Q = [self \in Nodes \mapsto \{\}]$
$\land\ pc = [self \in ProcSet \mapsto \text{“P”}]$

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

$PS(self) \triangleq$
$\land\ pc[self] = \text{“PS”}$
$\land$ IF $pt[self] \leq counter$
THEN $\land$ IF $pt[self] = 0$
THEN $\land\ Q' = [Q \text{ EXCEPT } ![self] = Nodes]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“L1”}]$
ELSE $\land\ (\forall\, i \in Nodes : pt[i] \geq 1 \lor \neg up[i])$
$\land\ Q' = [Q \text{ EXCEPT } ![self] = Nodes]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“L2”}]$
ELSE $\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“PR”}]$
$\land\ Q' = Q$
$\land$ UNCHANGED $\langle FailNum, up, pt, t, d, mb, v\rangle$

$L3(self) \triangleq$
$\land\ pc[self] = \text{“L3”}$
$\land\ pt' = [pt \text{ EXCEPT } ![self] = pt[self] + 1]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“PS”}]$
$\land$ UNCHANGED $\langle FailNum, up, t, d, mb, v, Q\rangle$

$L1(self) \triangleq$
$\land\ pc[self] = \text{“L1”}$
$\land$ IF $up[self] \land Q[self] \neq \{\}$
THEN $\land\ \exists\, p \in Q[self] :$
$\land$ IF $FailNum > 0 \land up[self]$
THEN $\land\ \lor\ \land\ up' = [up \text{ EXCEPT } ![self] = \text{FALSE}]$
$\land\ FailNum' = FailNum - 1$
$\land\ counter' = counter + 1$
$\lor\ \land$ TRUE
$\land$ UNCHANGED $\langle FailNum, up\rangle$
ELSE $\land$ TRUE

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land \text{ UNCHANGED } \langle \mathit{FailNum},\ \mathit{up} \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad \land \text{ IF } \neg \mathit{up}'[\mathit{self}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{THEN } \land\ t' = [t \text{ EXCEPT } ![\mathit{self}] = \text{TRUE}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \land\ \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“PR”}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \land \text{ UNCHANGED } \langle \mathit{mb},\ Q \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{ELSE } \land\ \mathit{mb}' = [\mathit{mb} \text{ EXCEPT } ![p] = \mathit{mb}[p] \cup \{v[\mathit{self}]\}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \land\ Q' = [Q \text{ EXCEPT } ![\mathit{self}] = Q[\mathit{self}] \setminus \{p\}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \land\ \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“L1”}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \land\ t' = t \\
\qquad\qquad\qquad\qquad\quad \text{ELSE } \land\ \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“L3”}] \\
\qquad\qquad\qquad\qquad\qquad\quad\ \land \text{ UNCHANGED } \langle \mathit{FailNum},\ \mathit{up},\ t,\ \mathit{mb},\ Q \rangle \\
\qquad\qquad\qquad \land \text{ UNCHANGED } \langle \mathit{pt},\ d,\ v \rangle \\
\\
L2(\mathit{self}) \triangleq\ \land\ \mathit{pc}[\mathit{self}] = \text{“L2”} \\
\qquad\qquad\qquad \land \text{ IF } \mathit{up}[\mathit{self}] \land Q[\mathit{self}] \neq \{\} \\
\qquad\qquad\qquad\quad \text{THEN } \land \exists\, p \in Q[\mathit{self}] : \\
\qquad\qquad\qquad\qquad\qquad \land \text{ IF } \mathit{FailNum} > 0 \land \mathit{up}[\mathit{self}] \\
\qquad\qquad\qquad\qquad\qquad\quad \text{THEN } \land\ \lor\ \land\ \mathit{up}' = [\mathit{up} \text{ EXCEPT } ![\mathit{self}] = \text{FALSE}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ \mathit{FailNum}' = \mathit{FailNum} - 1 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ \mathit{counter}' \ = \mathit{counter} + 1 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \lor\ \land \text{ TRUE} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \text{ UNCHANGED } \langle \mathit{FailNum},\ \mathit{up} \rangle \\
\qquad\qquad\qquad\qquad\qquad\quad \text{ELSE } \land \text{ TRUE} \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land \text{ UNCHANGED } \langle \mathit{FailNum},\ \mathit{up} \rangle \\
\qquad\qquad\qquad\qquad\qquad \land \text{ IF } \neg \mathit{up}'[\mathit{self}] \\
\qquad\qquad\qquad\qquad\qquad\quad \text{THEN } \land\ t' = [t \text{ EXCEPT } ![\mathit{self}] = \text{TRUE}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“PR”}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land \text{ UNCHANGED } \langle \mathit{mb},\ Q \rangle \\
\qquad\qquad\qquad\qquad\qquad\quad \text{ELSE } \land\ \mathit{mb}' = [\mathit{mb} \text{ EXCEPT } ![p] = \mathit{mb}[p] \cup \mathit{mb}[\mathit{self}]] \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ Q' = [Q \text{ EXCEPT } ![\mathit{self}] = Q[\mathit{self}] \setminus \{p\}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“L2”}] \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ t' = t \\
\qquad\qquad\qquad\qquad\quad \text{ELSE } \land\ \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“L3”}] \\
\qquad\qquad\qquad\qquad\qquad\quad\ \land \text{ UNCHANGED } \langle \mathit{FailNum},\ \mathit{up},\ t,\ \mathit{mb},\ Q \rangle \\
\qquad\qquad\qquad \land \text{ UNCHANGED } \langle \mathit{pt},\ d,\ v \rangle \\
\\
PR(\mathit{self}) \triangleq\ \land\ \mathit{pc}[\mathit{self}] = \text{“PR”} \\
\qquad\qquad\qquad \land\ (\forall\, n \in \mathit{Nodes} : (\mathit{pt}[n] = \mathit{counter} + 1 \lor \mathit{up}[n] = \text{FALSE})) \\
\qquad\qquad\qquad \land \text{ IF } \mathit{up}[\mathit{self}] \\
\qquad\qquad\qquad\quad \text{THEN } \land\ d' = [d \text{ EXCEPT } ![\mathit{self}] = \mathit{SetMin}(\mathit{mb}[\mathit{self}])] \\
\qquad\qquad\qquad\quad \text{ELSE } \land \text{ TRUE} \\
\qquad\qquad\qquad\qquad\ \land\ d' = d \\
\qquad\qquad\qquad \land\ t' = [t \text{ EXCEPT } ![\mathit{self}] = \text{TRUE}] \\
\qquad\qquad\qquad \land\ \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“Done”}] \\
\qquad\qquad\qquad \land \text{ UNCHANGED } \langle \mathit{FailNum},\ \mathit{up},\ \mathit{pt},\ \mathit{mb},\ v,\ Q \rangle
\end{array}
$$

$n(self) \triangleq P(self) \lor PS(self) \lor L3(self) \lor L1(self) \lor L2(self)$
$\qquad\qquad \lor PR(self)$

$Next \triangleq (\exists\, self \in Nodes : n(self))$
$\qquad \lor$ Disjunct to prevent deadlock on termination
$\qquad\quad ((\forall\, self \in ProcSet : pc[self] = \text{“Done”}) \land \text{UNCHANGED } vars)$

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

$Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})$

END TRANSLATION

$Inv \triangleq (\exists\, i \in Nodes : \neg t[i]) \lor (\forall\, l,\, m \in Nodes : \neg up[l] \lor \neg up[m] \lor d[l] = d[m])$

---

* Modification History
* Last modified *Tue Oct* 24 15:27:38 *EDT* 2017 by *xinboyu*
* Last modified *Tue Oct* 24 15:17:22 *EDT* 2017 by kz-*pc*
* Created *Tue Oct* 24 14:42:35 *EDT* 2017 by kz-*pc*