MODULE *BakeryDistributed*

This is the *PlusCal* specification of the distributed bakery algorithm in the paper

Deconstructing the Bakery to Build a Distributed State Machine

We assume here that you have read the *BakeryDeconstructed* specification, whose comments explain the structure of this *PlusCal* translation of the pseudo-code in the paper, and how it was model checked.

The statements in gray in the paper's pseudo-code, which involve the unnecessary variable *localCh*, are identified here by lines that end with the comment ERASABLE in the *PlusCal* code and where *localCh* appears in the *TypeOK* invariant.

EXTENDS *Integers*, *Sequences*

$$q \ll r \triangleq \lor\ q[1] < r[1] \lor\ \land\ q[1] = r[1] \land\ q[2] < r[2]$$

CONSTANT N
ASSUME $N \in Nat \setminus \{0\}$

$Nodes \triangleq 1 \,..\, N$
$OtherNodes(i) \triangleq Nodes \setminus \{i\}$
$ProcIds \triangleq \{\langle i \rangle : i \in Nodes\}$
$SubProcs \triangleq \{p \in Nodes \times Nodes : p[1] \neq p[2]\}$
$MsgProcs \triangleq \{p \in Nodes \times Nodes \times \{\text{“msg”}\} : p[1] \neq p[2]\}$
$SubProcsOf(i) \triangleq \{p \in SubProcs : p[1] = i\}$

$ack \triangleq$ CHOOSE $v : v \notin Nat$

****

```
--algorithm Decon{
  variables number = [i ∈ Nodes ↦ 0],
            localNum = [i ∈ Nodes ↦ [j ∈ OtherNodes(i) ↦ 0]],
            localCh  = [i ∈ Nodes ↦ [j ∈ OtherNodes(i) ↦ 0]],    ERASABLE
            ackRcvd  = [i ∈ Nodes ↦ [j ∈ OtherNodes(i) ↦ 0]],
            q = [i ∈ Nodes ↦ [j ∈ OtherNodes(i) ↦ ⟨⟩]]

  fair process (main ∈ ProcIds){
    ncs:- while (TRUE){
            skip ;  noncritical section
        M: await ∀ p ∈ SubProcsOf(self[1]) : pc[p] = "L0" ;
            with (v ∈ {n ∈ Nat \ {0} :
                          ∀ j ∈ OtherNodes(self[1]) :
                            n > localNum[self[1]][j]}){
              number[self[1]] := v ;
              q[self[1]] := [j ∈ OtherNodes(self[1])
                              ↦ Append(q[self[1]][j], v)]
            } ;
```

```
        L:  await ∀ p ∈ SubProcsOf(self[1]) : pc[p] = "ch" ;
       cs:  skip ;   critical section
        P:  ackRcvd[self[1]] := [j ∈ OtherNodes(self[1]) ↦ 0] ;
            number[self[1]] := 0 ;
            q[self[1]] := [j ∈ OtherNodes(self[1])
                           ↦ Append(q[self[1]][j], 0)]
         }
  }

  fair process (sub ∈ SubProcs){
    ch: while (TRUE){
          await pc[⟨self[1]⟩] = "M" ;
          localCh[self[2]][self[1]] := 1 ;    ERASABLE
    L0:   await pc[⟨self[1]⟩] = "L" ;
          await ackRcvd[self[1]][self[2]] = 1 ;
          localCh[self[2]][self[1]] := 0 ;          ERASABLE
    L2:   await localCh[self[1]][self[2]] = 0 ;   ERASABLE
    L3:   await  ∨ localNum[self[1]][self[2]] = 0
                 ∨ ⟨number[self[1]], self[1]⟩ ≪
                    ⟨localNum[self[1]][self[2]], self[2]⟩
        }
  }

  fair process (msg ∈ MsgProcs){
    wr: while (TRUE){
          await q[self[2]][self[1]] ≠ ⟨⟩ ;
          with (v = Head(q[self[2]][self[1]])){
            if (v   = ack){ackRcvd[self[1]][self[2]] := 1}
            else {localNum[self[1]][self[2]] := v} ;
            if (v ∈ {0, ack}){
               q[self[2]][self[1]] := Tail(q[self[2]][self[1]])}
            else {q[self[2]][self[1]] := Tail(q[self[2]][self[1]]) ||
                  q[self[1]][self[2]] := Append(q[self[1]][self[2]], ack)}
            }
        }
  }
}
```

****

BEGIN TRANSLATION ($chksum(pcal) =$ "$d4d60f14$" $\wedge\ chksum(tla) =$ "$8b3daef$")

VARIABLES $number$, $localNum$, $localCh$, $ackRcvd$, q, pc

$vars \triangleq \langle number,\ localNum,\ localCh,\ ackRcvd,\ q,\ pc\rangle$

$ProcSet \triangleq (ProcIds) \cup (SubProcs) \cup (MsgProcs)$

$Init \triangleq$ Global variables

$\land\ number = [i \in Nodes \mapsto 0]$
$\land\ localNum = [i \in Nodes \mapsto [j \in OtherNodes(i) \mapsto 0]]$
$\land\ localCh\ \ = [i \in Nodes \mapsto [j \in OtherNodes(i) \mapsto 0]]$
$\land\ ackRcvd = [i \in Nodes \mapsto [j \in OtherNodes(i) \mapsto 0]]$
$\land\ q = [i \in Nodes \mapsto [j \in OtherNodes(i) \mapsto \langle\rangle]]$
$\land\ pc = [self \in ProcSet \mapsto \text{CASE } self \in ProcIds \to \text{“ncs”}$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \square\ \ self \in SubProcs\ \to \text{“ch”}$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \square\ \ self \in MsgProcs \to \text{“wr”}]$

$ncs(self) \triangleq \land\ pc[self] = \text{“ncs”}$
$\qquad\qquad\land\ \text{TRUE}$
$\qquad\qquad\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“M”}]$
$\qquad\qquad\land\ \text{UNCHANGED}\ \langle number,\ localNum,\ localCh,\ ackRcvd,\ q\rangle$

$M(self) \triangleq \land\ pc[self] = \text{“M”}$
$\qquad\qquad\land\ \forall\, p \in SubProcsOf(self[1]) : pc[p] = \text{“L0”}$
$\qquad\qquad\land\ \exists\, v \in \{n \in Nat \setminus \{0\} :$
$\qquad\qquad\qquad\qquad \forall\, j \in OtherNodes(self[1])\ \ :$
$\qquad\qquad\qquad\qquad\quad n > localNum[self[1]][j]\} :$
$\qquad\qquad\qquad\land\ number' = [number\ \text{EXCEPT}\ ![self[1]] = v]$
$\qquad\qquad\qquad\land\ q' = [q\ \text{EXCEPT}\ ![self[1]] = [j \in OtherNodes(self[1])$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mapsto Append(q[self[1]][j],\ v)]]$
$\qquad\qquad\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“L”}]$
$\qquad\qquad\land\ \text{UNCHANGED}\ \langle localNum,\ localCh,\ ackRcvd\rangle$

$L(self) \triangleq \land\ pc[self] = \text{“L”}$
$\qquad\qquad\land\ \forall\, p \in SubProcsOf(self[1]) : pc[p] = \text{“ch”}$
$\qquad\qquad\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“cs”}]$
$\qquad\qquad\land\ \text{UNCHANGED}\ \langle number,\ localNum,\ localCh,\ ackRcvd,\ q\rangle$

$cs(self) \triangleq \land\ pc[self] = \text{“cs”}$
$\qquad\qquad\land\ \text{TRUE}$
$\qquad\qquad\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“P”}]$
$\qquad\qquad\land\ \text{UNCHANGED}\ \langle number,\ localNum,\ localCh,\ ackRcvd,\ q\rangle$

$P(self) \triangleq \land\ pc[self]\ \ = \text{“P”}$
$\qquad\qquad\land\ ackRcvd' = [ackRcvd\ \text{EXCEPT}\ ![self[1]] = [j \in OtherNodes(self[1]) \mapsto 0]]$
$\qquad\qquad\land\ number' = [number\ \text{EXCEPT}\ ![self[1]] = 0]$
$\qquad\qquad\land\ q' = [q\ \text{EXCEPT}\ ![self[1]] = [j \in OtherNodes(self[1])$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mapsto Append(q[self[1]][j],\ 0)]]$
$\qquad\qquad\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“ncs”}]$
$\qquad\qquad\land\ \text{UNCHANGED}\ \langle localNum,\ localCh\rangle$

$main(self) \triangleq\ ncs(self) \lor M(self) \lor L(self) \lor cs(self) \lor P(self)$

$ch(self) \triangleq \land\ pc[self] = \text{“ch”}$
$\qquad\qquad\land\ pc[\langle self[1]\rangle] = \text{“M”}$

$\land\ localCh' = [localCh \text{ EXCEPT } ![self[2]][self[1]] = 1]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“L0”}]$
$\land \text{ UNCHANGED } \langle number,\ localNum,\ ackRcvd,\ q\rangle$

$L0(self) \triangleq \land\ pc[self] = \text{“L0”}$
$\land\ pc[\langle self[1]\rangle] = \text{“L”}$
$\land\ ackRcvd[self[1]][self[2]] = 1$
$\land\ localCh' = [localCh \text{ EXCEPT } ![self[2]][self[1]] = 0]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“L2”}]$
$\land \text{ UNCHANGED } \langle number,\ localNum,\ ackRcvd,\ q\rangle$

$L2(self) \triangleq \land\ pc[self] = \text{“L2”}$
$\land\ localCh[self[1]][self[2]] = 0$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“L3”}]$
$\land \text{ UNCHANGED } \langle number,\ localNum,\ localCh,\ ackRcvd,\ q\rangle$

$L3(self) \triangleq \land\ pc[self] = \text{“L3”}$
$\land\ \lor\ localNum[self[1]][self[2]] = 0$
$\quad \lor \langle number[self[1]],\ self[1]\rangle \ll$
$\qquad \langle localNum[self[1]][self[2]],\ self[2]\rangle$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“ch”}]$
$\land \text{ UNCHANGED } \langle number,\ localNum,\ localCh,\ ackRcvd,\ q\rangle$

$sub(self) \triangleq ch(self) \lor L0(self) \lor L2(self) \lor L3(self)$

$wr(self) \triangleq \land\ pc[self] = \text{“wr”}$
$\land\ q[self[2]][self[1]] \neq \langle\rangle$
$\land \text{ LET } v \triangleq Head(q[self[2]][self[1]]) \text{ IN}$
$\quad \land \text{ IF } v = ack$
$\qquad \text{THEN } \land\ ackRcvd' = [ackRcvd \text{ EXCEPT } ![self[1]][self[2]] = 1]$
$\qquad\quad \land \text{ UNCHANGED } localNum$
$\qquad \text{ELSE } \land\ localNum' = [localNum \text{ EXCEPT } ![self[1]][self[2]] = v]$
$\qquad\quad \land \text{ UNCHANGED } ackRcvd$
$\quad \land \text{ IF } v \in \{0,\ ack\}$
$\qquad \text{THEN } \land\ q' = [q \text{ EXCEPT } ![self[2]][self[1]] = Tail(q[self[2]][self[1]])]$
$\qquad \text{ELSE } \land\ q' = [q \text{ EXCEPT } ![self[2]][self[1]] = Tail(q[self[2]][self[1]]),$
$\qquad\qquad\qquad\qquad\qquad\quad ![self[1]][self[2]] = Append(q[self[1]][self[2]],\ ack)]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“wr”}]$
$\land \text{ UNCHANGED } \langle number,\ localCh\rangle$

$msg(self) \triangleq wr(self)$

$Next \triangleq (\exists\, self \in ProcIds : main(self))$
$\quad \lor (\exists\, self \in SubProcs : sub(self))$
$\quad \lor (\exists\, self \in MsgProcs : msg(self))$

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

$\land \forall\, self \in ProcIds : \mathrm{WF}_{vars}((pc[self] \neq \text{“ncs”}) \land main(self))$
$\land \forall\, self \in SubProcs : \mathrm{WF}_{vars}(sub(self))$
$\land \forall\, self \in MsgProcs : \mathrm{WF}_{vars}(msg(self))$

END TRANSLATION

$TypeOK \triangleq$
$\land\ number \in [Nodes \to Nat]$
$\land\ \land$ DOMAIN $localNum = Nodes$
$\quad \land \forall\, i \in Nodes : localNum[i] \in [OtherNodes(i) \to Nat]$
$\land\ \land$ DOMAIN $localCh = Nodes$ ERASABLE
$\quad \land \forall\, i \in Nodes : localCh[i] \in [OtherNodes(i) \to \{0, 1\}]$ ERASABLE
$\land\ \land$ DOMAIN $ackRcvd = Nodes$
$\quad \land \forall\, i \in Nodes : ackRcvd[i] \in [OtherNodes(i) \to \{0, 1\}]$
$\land\ \land$ DOMAIN $q = Nodes$
$\quad \land \forall\, i \in Nodes : q[i] \in [OtherNodes(i) \to Seq(Nat \cup \{ack\})]$
$\land\ \land$ DOMAIN $pc = ProcSet$
$\quad \land \forall\, p \in ProcSet :$
$\qquad$ CASE $p \in ProcIds \to pc[p] \in \{\text{“ncs”}, \text{“M”}, \text{“L”}, \text{“cs”}, \text{“P”}\}$
$\qquad\quad \square \quad p \in SubProcs \to pc[p] \in \{\text{“ch”}, \text{“L0”}, \text{“L2”}, \text{“L3”}\}$
$\qquad\quad \square \quad p \in MsgProcs \to pc[p] = \text{“wr”}$

$MutualExclusion \triangleq \forall\, p, r \in ProcIds : (p \neq r) \Rightarrow (\{pc[p], pc[r]\} \neq \{\text{“cs”}\})$

$StarvationFree \triangleq \forall\, p \in ProcIds : (pc[p] = \text{“M”}) \rightsquigarrow (pc[p] = \text{“cs”})$

$TestMaxNum \triangleq 6$
$TestNat \triangleq 0\,..\,(TestMaxNum + 1)$

TEST RESULTS

TLC has tested that *TypeOK* and *MutualExclusion* are invariants of the algorithm, and that the algorithm satisfies the temporal property *StarvationFree*. As a sanity check, some smaller models were used to check that, if fairness is not disabled for the *ncs* action, then the algorithm satisfies the following property, which asserts that every process executes the critical section infinitely many times.

$\forall\, i \in Procs : \Box\Diamond(pc[\langle i \rangle] = \text{“cs”})$

The largest model that was tested was for $N = 3$ and $TestMaxNum = 6$. It had 24,943,042 reachable states and was executed in a little less than 52 minutes on a 64-core machine using 55 worker threads.

---

* Modification History
* Last modified *Mon Aug* 02 15:23:28 *PDT* 2021 by *lamport*
* Created *Tue Apr* 27 10:33:38 *PDT* 2021 by *lamport*