MODULE *Bakery*

The bakery algorithm originally appeared in:

Leslie *Lamport* A New Solution of *Dijkstra*'s Concurrent Programming Problem Communications of the *ACM* 17, 8 (August 1974), 453-455

The code for the algorithm given in that paper is :

```
begin integer j;
L1: choosing [i] := 1 ;
    number[i] := 1 + maximum (number[1],..., number[N]);
    choosing[i] := 0;
    for j = 1 step l until N do
       begin
         L2: if choosing[j] /= 0 then goto L2;
         L3: if number[j] /= 0 and (number [j], j) < (number[i],i)
               then goto L3;
       end;
    critical section;
    number[i] := 0;
    noncritical section;
    goto L1 ;
end
```

This *PlusCal* version of the Atomic *Bakery* algorithm is one in which variables whose initial values are not used are initialized to particular type-correct values. If the variables were left uninitialized, the *PlusCal* translation would initialize them to a particular unspecified value. This would complicate the proof because it would make the type-correctness invariant more complicated, but it would be efficient to model check. We could write a version that is more elegant and easy to prove, but less efficient to model check, by initializing the variables to arbitrarily chosen type-correct values.

EXTENDS *Naturals*, *TLAPS*

We first declare $N$ to be the number of processes, and we assume that $N$ is a natural number.

CONSTANT $N$

ASSUME $N \in Nat$

We define *Procs* to be the set $\{1, 2, \ldots, N\}$ of processes.

$Procs \triangleq 1 \,..\, N$

$\prec$ is defined to be the lexicographical less-than relation on pairs of numbers.

$$a \prec b \triangleq \lor\, a[1] < b[1] \\ \lor\, (a[1] = b[1]) \land (a[2] < b[2])$$

```
 **      this is a comment containing the PlusCal code *

--algorithm Bakery
{variables num = [i ∈ Procs ↦ 0], flag = [i ∈ Procs ↦ FALSE];
 fair process(p ∈ Procs)
   variables unchecked = {}, max = 0, nxt = 1 ;
   {ncs : - while(TRUE)
        {e1 : either{flag[self] := ¬flag[self];
                     goto e1}
             or     {flag[self] := TRUE;
```

```
                    unchecked := Procs \ {self};
                    max := 0
                  };
          e2 : while(unchecked ≠ {})
                {with(i ∈ unchecked)
                  {unchecked := unchecked \ {i};
                   if(num[i] > max){max := num[i]}
                  }
                };
          e3 : either{with(k ∈ Nat){num[self] := k};
                      goto e3}
               or     {with(i ∈ {j ∈ Nat : j > max})
                        {num[self] := i}
                      };
          e4 : either{flag[self] := ¬flag[self];
                      goto e4}
               or     {flag[self] := FALSE;
                      unchecked := Procs \ {self}
                      };
          w1 : while(unchecked ≠ {})
               {     with(i ∈ unchecked){nxt := i};
                     await ¬flag[nxt];
                w2 : await ∨ num[nxt] = 0
                           ∨ ⟨num[self], self⟩ ≺ ⟨num[nxt], nxt⟩;
                     unchecked := unchecked \ {nxt};
               };
          cs : skip ; \ * the critical section;
          exit : either{with(k ∈ Nat){num[self] := k};
                        goto exit}
               or     {num[self] := 0}
        }
   }
}

   this ends the comment containing the PlusCal code
***********
```

BEGIN TRANSLATION (this begins the translation of the *PlusCal* code)

VARIABLES $num$, $flag$, $pc$, $unchecked$, $max$, $nxt$

$vars \triangleq \langle num, flag, pc, unchecked, max, nxt \rangle$

$ProcSet \triangleq (Procs)$

$Init \triangleq$ Global variables

$\land num = [i \in Procs \mapsto 0]$

$\land flag = [i \in Procs \mapsto \text{FALSE}]$

Process $p$

$\land unchecked = [self \in Procs \mapsto \{\}]$

$\land max = [self \in Procs \mapsto 0]$

$\land nxt = [self \in Procs \mapsto 1]$

$$\land\, pc = [self \in ProcSet \mapsto \text{“ncs”}]$$

$$\begin{array}{l}
ncs(self) \triangleq \land\, pc[self] = \text{“ncs”} \\
\qquad\qquad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e1”}] \\
\qquad\qquad \land \text{ UNCHANGED } \langle num,\, flag,\, unchecked,\, max,\, nxt\rangle
\end{array}$$

$$\begin{array}{l}
e1(self) \triangleq \land\, pc[self] = \text{“e1”} \\
\qquad\qquad \land\ \lor\ \land\, flag' = [flag \text{ EXCEPT } ![self] = \neg flag[self]] \\
\qquad\qquad\qquad\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e1”}] \\
\qquad\qquad\qquad\quad \land \text{ UNCHANGED } \langle unchecked,\, max\rangle \\
\qquad\qquad\qquad \lor\ \land\, flag' = [flag \text{ EXCEPT } ![self] = \text{TRUE}] \\
\qquad\qquad\qquad\quad \land\, unchecked' = [unchecked \text{ EXCEPT } ![self] = Procs \setminus \{self\}] \\
\qquad\qquad\qquad\quad \land\, max' = [max \text{ EXCEPT } ![self] = 0] \\
\qquad\qquad\qquad\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e2”}] \\
\qquad\qquad \land \text{ UNCHANGED } \langle num,\, nxt\rangle
\end{array}$$

$$\begin{array}{l}
e2(self) \triangleq \land\, pc[self] = \text{“e2”} \\
\qquad\qquad \land \text{ IF } unchecked[self] \neq \{\} \\
\qquad\qquad\qquad \text{THEN } \land \exists\, i \in unchecked[self] : \\
\qquad\qquad\qquad\qquad\qquad \land\, unchecked' = [unchecked \text{ EXCEPT } ![self] = unchecked[self] \setminus \{i\}] \\
\qquad\qquad\qquad\qquad\qquad \land \text{ IF } num[i] > max[self] \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{THEN } \land\, max' = [max \text{ EXCEPT } ![self] = num[i]] \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE } \land \text{ TRUE} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \land\, max' = max \\
\qquad\qquad\qquad\qquad\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e2”}] \\
\qquad\qquad\qquad \text{ELSE } \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e3”}] \\
\qquad\qquad\qquad\qquad\quad \land \text{ UNCHANGED } \langle unchecked,\, max\rangle \\
\qquad\qquad \land \text{ UNCHANGED } \langle num,\, flag,\, nxt\rangle
\end{array}$$

$$\begin{array}{l}
e3(self) \triangleq \land\, pc[self] = \text{“e3”} \\
\qquad\qquad \land\ \lor\ \land \exists\, k \in Nat : \\
\qquad\qquad\qquad\quad\ num' = [num \text{ EXCEPT } ![self] = k] \\
\qquad\qquad\qquad\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e3”}] \\
\qquad\qquad\qquad \lor\ \land \exists\, i \in \{j \in Nat : j > max[self]\} : \\
\qquad\qquad\qquad\quad\ num' = [num \text{ EXCEPT } ![self] = i] \\
\qquad\qquad\qquad\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e4”}] \\
\qquad\qquad \land \text{ UNCHANGED } \langle flag,\, unchecked,\, max,\, nxt\rangle
\end{array}$$

$$\begin{array}{l}
e4(self) \triangleq \land\, pc[self] = \text{“e4”} \\
\qquad\qquad \land\ \lor\ \land\, flag' = [flag \text{ EXCEPT } ![self] = \neg flag[self]] \\
\qquad\qquad\qquad\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“e4”}] \\
\qquad\qquad\qquad\quad \land \text{ UNCHANGED } unchecked \\
\qquad\qquad\qquad \lor\ \land\, flag' = [flag \text{ EXCEPT } ![self] = \text{FALSE}] \\
\qquad\qquad\qquad\quad \land\, unchecked' = [unchecked \text{ EXCEPT } ![self] = Procs \setminus \{self\}] \\
\qquad\qquad\qquad\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“w1”}] \\
\qquad\qquad \land \text{ UNCHANGED } \langle num,\, max,\, nxt\rangle
\end{array}$$

$$
\begin{array}{l}
w1(self) \triangleq \land\ pc[self] = \text{“w1”} \\
\qquad \land\ \text{IF } unchecked[self] \neq \{\} \\
\qquad\qquad \text{THEN } \land\ \exists\, i \in unchecked[self] : \\
\qquad\qquad\qquad\qquad nxt' = [nxt \text{ EXCEPT } ![self] = i] \\
\qquad\qquad\qquad\ \land\ \neg flag[nxt'[self]] \\
\qquad\qquad\qquad\ \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“w2”}] \\
\qquad\qquad \text{ELSE } \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“cs”}] \\
\qquad\qquad\qquad\ \land\ nxt' = nxt \\
\qquad \land\ \text{UNCHANGED } \langle num,\, flag,\, unchecked,\, max \rangle
\end{array}
$$

$$
\begin{array}{l}
w2(self) \triangleq \land\ pc[self] = \text{“w2”} \\
\qquad \land\ \lor\ num[nxt[self]] = 0 \\
\qquad\quad \lor\ \langle num[self],\, self \rangle \prec \langle num[nxt[self]],\, nxt[self] \rangle \\
\qquad \land\ unchecked' = [unchecked \text{ EXCEPT } ![self] = unchecked[self] \setminus \{nxt[self]\}] \\
\qquad \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“w1”}] \\
\qquad \land\ \text{UNCHANGED } \langle num,\, flag,\, max,\, nxt \rangle
\end{array}
$$

$$
\begin{array}{l}
cs(self) \triangleq \land\ pc[self] = \text{“cs”} \\
\qquad \land\ \text{TRUE} \\
\qquad \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“exit”}] \\
\qquad \land\ \text{UNCHANGED } \langle num,\, flag,\, unchecked,\, max,\, nxt \rangle
\end{array}
$$

$$
\begin{array}{l}
exit(self) \triangleq \land\ pc[self] = \text{“exit”} \\
\qquad \land\ \lor\ \land\ \exists\, k \in Nat : \\
\qquad\qquad\qquad\qquad num' = [num \text{ EXCEPT } ![self] = k] \\
\qquad\qquad\qquad\ \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“exit”}] \\
\qquad\qquad\ \lor\ \land\ num' = [num \text{ EXCEPT } ![self] = 0] \\
\qquad\qquad\qquad\ \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“ncs”}] \\
\qquad \land\ \text{UNCHANGED } \langle flag,\, unchecked,\, max,\, nxt \rangle
\end{array}
$$

$$
\begin{array}{l}
p(self) \triangleq\ ncs(self) \lor e1(self) \lor e2(self) \lor e3(self) \lor e4(self) \\
\qquad\qquad \lor w1(self) \lor w2(self) \lor cs(self) \lor exit(self)
\end{array}
$$

$$
Next \triangleq\ (\exists\, self \in Procs : p(self))
$$

$$
\begin{array}{l}
Spec \triangleq\ \land\ Init \land \Box[Next]_{vars} \\
\qquad\qquad \land\ \forall\, self \in Procs : \mathrm{WF}_{vars}((pc[self] \neq \text{“ncs”}) \land p(self))
\end{array}
$$

END TRANSLATION (this ends the translation of the *PlusCal* code)

*MutualExclusion* asserts that no two distinct processes are in their critical sections.

$$
\begin{array}{l}
MutualExclusion \triangleq\ \forall\, i,\, j \in Procs : (i \neq j) \Rightarrow \neg\, \land\ pc[i] = \text{“cs”} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land\ pc[j] = \text{“cs”}
\end{array}
$$

---

The Inductive Invariant

*TypeOK* is the type-correctness invariant.

$$
\begin{array}{lll}
\mathit{TypeOK} & \triangleq & \land\ \mathit{num} \in [\mathit{Procs} \to \mathit{Nat}] \\
& & \land\ \mathit{flag} \in [\mathit{Procs} \to \text{BOOLEAN }] \\
& & \land\ \mathit{unchecked} \in [\mathit{Procs} \to \text{SUBSET } \mathit{Procs}] \\
& & \land\ \mathit{max} \in [\mathit{Procs} \to \mathit{Nat}] \\
& & \land\ \mathit{nxt}\ \in [\mathit{Procs} \to \mathit{Procs}] \\
& & \land\ \mathit{pc} \in [\mathit{Procs} \to \{\text{“ncs”},\ \text{“e1”},\ \text{“e2”},\ \text{“e3”}, \\
& & \qquad\qquad \text{“e4”},\ \text{“w1”},\ \text{“w2”},\ \text{“cs”},\ \text{“exit”}\}]
\end{array}
$$

*Before*$(i, j)$ is a condition that implies that $\mathit{num}[i] > 0$ and, if $j$ is trying to enter its critical section and $i$ does not change $\mathit{num}[i]$, then $j$ either has or will choose a value of $\mathit{num}[j]$ for which

$\langle \mathit{num}[i],\ i\rangle \prec \langle \mathit{num}[j],\ j\rangle$

is true.

$$
\begin{array}{lll}
\mathit{Before}(i,\ j) & \triangleq & \land\ \mathit{num}[i] > 0 \\
& & \land\ \lor\ \mathit{pc}[j] \in \{\text{“ncs”},\ \text{“e1”},\ \text{“exit”}\} \\
& & \quad \lor\ \land\ \mathit{pc}[j] = \text{“e2”} \\
& & \qquad \land\ \lor\ i \in \mathit{unchecked}[j] \\
& & \qquad\quad \lor\ \mathit{max}[j] \geq \mathit{num}[i] \\
& & \quad \lor\ \land\ \mathit{pc}[j] = \text{“e3”} \\
& & \qquad \land\ \mathit{max}[j] \geq \mathit{num}[i] \\
& & \quad \lor\ \land\ \mathit{pc}[j] \in \{\text{“e4”},\ \text{“w1”},\ \text{“w2”}\} \\
& & \qquad \land\ \langle \mathit{num}[i],\ i\rangle \prec \langle \mathit{num}[j],\ j\rangle \\
& & \qquad \land\ (\mathit{pc}[j] \in \{\text{“w1”},\ \text{“w2”}\}) \Rightarrow (i \in \mathit{unchecked}[j])
\end{array}
$$

*Inv* is the complete inductive invariant.

$$
\begin{array}{lll}
\mathit{Inv} & \triangleq & \land\ \mathit{TypeOK} \\
& & \land\ \forall\, i \in \mathit{Procs} : \\
& & \quad \land\ (\mathit{pc}[i] \in \{\text{“ncs”},\ \text{“e1”},\ \text{“e2”}\})\ \Rightarrow\ (\mathit{num}[i] = 0) \\
& & \qquad \land\ (\mathit{pc}[i] \in \{\text{“e4”},\ \text{“w1”},\ \text{“w2”},\ \text{“cs”}\}) \Rightarrow (\mathit{num}[i] \neq 0) \\
& & \qquad \land\ (\mathit{pc}[i] \in \{\text{“e2”},\ \text{“e3”}\}) \Rightarrow \mathit{flag}[i] \\
& & \qquad \land\ (\mathit{pc}[i] = \text{“w2”}) \Rightarrow (\mathit{nxt}[i] \neq i) \\
& & \qquad \land\ \mathit{pc}[i] \in \{\ \text{“e2”},\text{“w1”},\ \text{“w2”}\} \Rightarrow i \notin \mathit{unchecked}[i] \\
& & \qquad \land\ (\mathit{pc}[i] \in \{\text{“w1”},\ \text{“w2”}\}) \Rightarrow \\
& & \qquad\quad \forall\, j \in (\mathit{Procs} \setminus \mathit{unchecked}[i]) \setminus \{i\} : \mathit{Before}(i,\ j) \\
& & \qquad \land\ \land\ (\mathit{pc}[i] = \text{“w2”}) \\
& & \qquad\quad \land\ \lor\ (\mathit{pc}[\mathit{nxt}[i]] = \text{“e2”}) \land (i \notin \mathit{unchecked}[\mathit{nxt}[i]]) \\
& & \qquad\quad\quad \lor\ \mathit{pc}[\mathit{nxt}[i]] = \text{“e3”} \\
& & \qquad\quad \Rightarrow \mathit{max}[\mathit{nxt}[i]] \geq \mathit{num}[i] \\
& & \qquad \land\ (\mathit{pc}[i] = \text{“cs”}) \Rightarrow \forall\, j \in \mathit{Procs} \setminus \{i\} : \mathit{Before}(i,\ j)
\end{array}
$$

---

Proof of Mutual Exclusion

This is a standard invariance proof, where $< 1 > 2$ asserts that any step of the algorithm (including a stuttering step) starting in a state in which *Inv* is true leaves *Inv* true. Step $< 1 > 4$ follows easily from $< 1 > 1$- $< 1 > 3$ by simple temporal reasoning.

THEOREM $Spec \Rightarrow \Box MutualExclusion$
$< 1 >$ $USEN \in NatDEFSProcs,\ TypeOK,\ Before,\ \prec,\ ProcSet$
$< 1 > 1. Init \Rightarrow Inv$
$BYDEFInit,\ Inv$
$< 1 > 2. Inv \wedge [Next]_{vars} \Rightarrow Inv'$
$< 2 > SUFFICES$ ASSUME $Inv,$
$[Next]_{vars}$
$PROVE\ Inv'$
$OBVIOUS$
$< 2 > 1.$ ASSUME $NEW self \in Procs,$
$ncs(self)$
$PROVE\ Inv'$
$BY < 2 > 1 DEF ncs,\ Inv$
$< 2 > 2.$ ASSUME $NEW self \in Procs,$
$e1(self)$
$PROVE\ Inv'$
$< 3 > . \wedge pc[self] =$ "e1"
$\wedge$ UNCHANGED $\langle num,\ nxt \rangle$
$BY < 2 > 2 DEF e1$
$< 3 > 1.$ CASE $\wedge flag' = [flag$ EXCEPT $![self] = \neg flag[self]]$
$\wedge pc' = [pc$ EXCEPT $![self] =$ "e1"$]$
$\wedge$ UNCHANGED $\langle unchecked,\ max \rangle$
$BY < 3 > 1 DEF Inv$
$< 3 > 2.$ CASE $\wedge flag' = [flag$ EXCEPT $![self] =$ TRUE$]$
$\wedge unchecked' = [unchecked$ EXCEPT $![self] = Procs \setminus \{self\}]$
$\wedge max' = [max$ EXCEPT $![self] = 0]$
$\wedge pc' = [pc$ EXCEPT $![self] =$ "e2"$]$
$BY < 3 > 2 DEF Inv$
$< 3 > .QED\ BY < 3 > 1,\ < 3 > 2,\ < 2 > 2 DEF e1$
$< 2 > 3.$ ASSUME $NEW self \in Procs,$
$e2(self)$
$PROVE\ Inv'$
$< 3 > . \wedge pc[self] =$ "e2"
$\wedge$ UNCHANGED $\langle num,\ flag,\ nxt \rangle$
$BY < 2 > 3 DEF e2$
$< 3 > 1.$ ASSUME $NEW i \in unchecked[self],$
$unchecked' = [unchecked$ EXCEPT $![self] = unchecked[self] \setminus \{i\}],$
$num[i] > max[self],$
$max' = [max$ EXCEPT $![self] = num[i]],$
$pc' = [pc$ EXCEPT $![self] =$ "e2"$]$
$PROVE\ Inv'$
$BY < 3 > 1,\ Z3T(10) DEF Inv$
$< 3 > 2.$ ASSUME $NEW i \in unchecked[self],$
$unchecked' = [unchecked$ EXCEPT $![self] = unchecked[self] \setminus \{i\}],$
$\neg(num[i] > max[self]),$

$max' = max$,
$pc' = [pc \text{ EXCEPT } ![self] = \text{“e2”}]$
$PROVE\ \ Inv'$
$< 4 > .\ TypeOK'\ \ BY < 3 > 2\ DEFInv$
$< 4 > 1. \forall\, ii \in Procs : (pc'[ii] \in \{\text{“e4”}, \text{“w1”}, \text{“w2”}, \text{“cs”}\}) \Rightarrow (num'[ii] \neq 0)$
$BY < 3 > 2\ DEFInv$
$< 4 > 2. \forall\, ii \in Procs : (pc'[ii] \in \{\text{“e2”}, \text{“e3”}\}) \Rightarrow flag'[ii]$
$BY < 3 > 2\ DEFInv$
$< 4 > 3. \forall\, ii \in Procs : (pc'[ii] = \text{“w2”}) \Rightarrow (nxt'[ii] \neq ii)$
$BY < 3 > 2\ DEFInv$
$< 4 > 4. \forall\, ii \in Procs : pc'[ii] \in \{\text{“e2”}, \text{“w1”}, \text{“w2”}\} \Rightarrow ii \notin unchecked'[ii]$
$BY < 3 > 2\ DEFInv$
$< 4 > 5. \forall\, ii \in Procs : (pc'[ii] \in \{\text{“w1”}, \text{“w2”}\}) \Rightarrow$
$\forall\, j \in (Procs \setminus unchecked'[ii]) \setminus \{ii\} : Before(ii, j)'$
$BY < 3 > 2\ DEFInv$
$< 4 > 6. \forall\, ii \in Procs :$
$\land\ (pc'[ii] = \text{“w2”})$
$\land\ \lor\ (pc'[nxt'[ii]] = \text{“e2”}) \land (ii \notin unchecked'[nxt'[ii]])$
$\lor\ pc'[nxt'[ii]] = \text{“e3”}$
$\Rightarrow max'[nxt'[ii]] \geq num'[ii]$
$BY < 3 > 2\ DEFInv$
$< 4 > 7. \forall\, ii \in Procs : (pc'[ii] = \text{“cs”}) \Rightarrow \forall\, j \in Procs \setminus \{ii\} : Before(ii, j)'$
$BY < 3 > 2\ DEFInv$
$< 4 > .\ QED\ \ BY\ \ < 4 > 0, < 4 > 1,\ < 4 > 2,\ < 4 > 3,\ < 4 > 4,\ < 4 > 5,\ < 4 > 6,\ < 4 > 7$
$< 3 > 3.\text{CASE}\ \land\ unchecked[self] = \{\}$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“e3”}]$
$\land\ \text{UNCHANGED}\ \langle unchecked, max\rangle$
$BY < 3 > 3\ DEFInv$
$< 3 > .\ QED\ \ BY < 3 > 1,\ < 3 > 2,\ < 3 > 3,\ < 2 > 3\ DEFe2$
$< 2 > 4.\text{ASSUME}\ NEW self \in Procs,$
$e3(self)$
$PROVE\ \ Inv'$
$< 3 > .\ \land\ pc[self] = \text{“e3”}$
$\land\ \text{UNCHANGED}\ \langle flag, unchecked, max, nxt\rangle$
$BY < 2 > 4\ DEFe3$
$< 3 > 1.\text{CASE}\ \land \exists\, k \in Nat :$
$num' = [num \text{ EXCEPT } ![self] = k]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“e3”}]$
$BY < 3 > 1\ DEFInv$
$< 3 > 2.\text{CASE}\ \land \exists\, i \in \{j \in Nat : j > max[self]\} :$
$num' = [num \text{ EXCEPT } ![self] = i]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“e4”}]$
$BY < 3 > 2,\ Z3\ DEFInv$
$< 3 > 3.\ QED\ \ BY < 3 > 1,\ < 3 > 2,\ < 2 > 4\ DEFe3$
$< 2 > 5.\text{ASSUME}\ NEW self \in Procs,$

$e4(self)$

*PROVE* $Inv'$

$< 3 >.$ $\land\ pc[self] =$ "e4"
$\land$ UNCHANGED $\langle num, max, nxt \rangle$

*BY* $< 2 > 5 DEFe4$

$< 3 > 1.$CASE $\land\ flag' = [flag$ EXCEPT $![self] = \neg flag[self]]$
$\land\ pc' = [pc$ EXCEPT $![self] =$ "e4"$]$
$\land$ UNCHANGED $unchecked$

*BY* $< 3 > 1 DEFInv$

$< 3 > 2.$CASE $\land\ flag' = [flag$ EXCEPT $![self] =$ FALSE$]$
$\land\ unchecked' = [unchecked$ EXCEPT $![self] = Procs \setminus \{self\}]$
$\land\ pc' = [pc$ EXCEPT $![self] =$ "w1"$]$

*BY* $< 3 > 2, Z3 T(30) DEFInv$

$< 3 >. QED$ *BY* $< 3 > 1,\ < 3 > 2,\ < 2 > 5 DEFe4$

$< 2 > 6.$ASSUME $NEWself \in Procs,$
$w1(self)$

*PROVE* $Inv'$

$< 3 >.$ $\land\ pc[self] =$ "w1"
$\land$ UNCHANGED $\langle num, flag, unchecked, max \rangle$

*BY* $< 2 > 6 DEFw1$

$< 3 > 1.$CASE $\land\ unchecked[self] \neq \{\}$
$\land \exists\, i \in unchecked[self] :$
$nxt' = [nxt$ EXCEPT $![self] = i]$
$\land \neg flag[nxt'[self]]$
$\land\ pc' = [pc$ EXCEPT $![self] =$ "w2"$]$

*BY* $< 3 > 1, Z3 DEFInv$

$< 3 > 2.$CASE $\land\ unchecked[self] = \{\}$
$\land\ pc' = [pc$ EXCEPT $![self] =$ "cs"$]$
$\land\ nxt' = nxt$

*BY* $< 3 > 2, Z3 DEFInv$

$< 3 >. QED$ *BY* $< 3 > 1,\ < 3 > 2,\ < 2 > 6 DEFw1$

$< 2 > 7.$ASSUME $NEWself \in Procs,$
$w2(self)$

*PROVE* $Inv'$

*BY* $< 2 > 7, Z3 DEFw2, Inv$

$< 2 > 8.$ASSUME $NEWself \in Procs,$
$cs(self)$

*PROVE* $Inv'$

*BY* $< 2 > 8, Z3 DEFcs, Inv$

$< 2 > 9.$ASSUME $NEWself \in Procs,$
$exit(self)$

*PROVE* $Inv'$

$< 3 >.$ $\land\ pc[self] =$ "exit"
$\land$ UNCHANGED $\langle flag, unchecked, max, nxt \rangle$

*BY* $< 2 > 9 DEFexit$

$< 3 > 1$.CASE $\land \exists\, k \in Nat :$
$num' = [num \text{ EXCEPT } ![self] = k]$
$\land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“exit”}]$
$BY < 3 > 1 DEFInv$
$< 3 > 2$.CASE $\land\, num' = [num \text{ EXCEPT } ![self] = 0]$
$\land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“ncs”}]$
$BY < 3 > 2 DEFInv$
$< 3 > .QED\ BY < 3 > 1,\ < 3 > 2,\ < 2 > 9 DEFexit$
$< 2 > 10$.CASE UNCHANGED $vars$
$BY < 2 > 10 DEFvars, Inv$
$< 2 > 11.QED$
$BY < 2 > 1,\ < 2 > 10,\ < 2 > 2,\ < 2 > 3,\ < 2 > 4,\ < 2 > 5,\ < 2 > 6,\ < 2 > 7,\ < 2 > 8,\ < 2 > 9$
$< 1 > 3. Inv \Rightarrow MutualExclusion$
$BY SMTDEF MutualExclusion, Inv$
$< 1 > 4.QED$
$BY < 1 > 1,\ < 1 > 2,\ < 1 > 3, PTLDEFSpec$

---

$Trying(i) \triangleq pc[i] = \text{“e1”}$
$InCS(i) \triangleq pc[i] = \text{“cs”}$
$DeadlockFree \triangleq (\exists\, i \in Procs : Trying(i)) \leadsto (\exists\, i \in Procs : InCS(i))$
$StarvationFree \triangleq \forall\, i \in Procs : Trying(i) \leadsto InCS(i)$

---

$II \triangleq \forall\, i \in Procs :$
$\land\ (pc[i] \in \{\text{“ncs”}, \text{“e1”}, \text{“e2”}\}) \Rightarrow (num[i] = 0)$ * not found Test 1 (21993 states)
$\land\, (pc[i] \in \{\text{“e4”}, \text{“w1”}, \text{“w2”}, \text{“cs”}\}) \Rightarrow (num[i] \neq 0)$ found Test 1
$\land\, (pc[i] \in \{\text{“e2”}, \text{“e3”}\}) \Rightarrow flag[i]$ found Test 1
$\land\, (pc[i] = \text{“w2”}) \Rightarrow (nxt[i] \neq i)$ not found Test 1 (12115 states) or with $N = 2$
$\land\, pc[i] \in \{\text{“e2”}, \text{“w1”}, \text{“w2”}\} \Rightarrow i \notin unchecked[i]$ found Test 1
$\land\, (pc[i] \in \{\text{“w1”}, \text{“w2”}\}) \Rightarrow$ found Test 1
$\forall\, j \in (Procs \setminus unchecked[i]) \setminus \{i\} : Before(i, j)$
$\land\ \land\, (pc[i] = \text{“w2”})$ found Test 1
$\land\ \lor\, (pc[nxt[i]] = \text{“e2”}) \land (i \notin unchecked[nxt[i]])$
$\lor\, pc[nxt[i]] = \text{“e3”}$
$\Rightarrow max[nxt[i]] \geq num[i]$
$\land\, (pc[i] = \text{“cs”}) \Rightarrow \forall\, j \in Procs \setminus \{i\} : Before(i, j)$ found Test 1

$IInit \triangleq \land\, num \in [Procs \to Nat]$
$\land\, flag \in [Procs \to \text{BOOLEAN }]$
$\land\, unchecked \in [Procs \to \text{SUBSET } Procs]$
$\land\, max \in [Procs \to\ Nat]$
$\land\, nxt\ \in [Procs \to\ Procs]$
$\land\, pc \in [Procs \to \{\text{“ncs”}, \text{“e1”}, \text{“e2”}, \text{“e3”},$
$\text{“e4”}, \text{“w1”}, \text{“w2”}, \text{“cs”}, \text{“exit”}\}]$
$\land\, II$

$ISpec \triangleq IInit \wedge \Box[Next]_{vars}$

* Modification History
* Last modified Sat *Mar* 07 08:41:02 *CET* 2020 by *merz*
* Last modified *Tue Aug* 27 12:23:10 *PDT* 2019 by *loki*
* Last modified Sat May 19 16:40:23 *CEST* 2018 by *merz*
* Last modified *Thu* May 17 07:02:45 *PDT* 2018 by *lamport*
* Created *Thu Nov* 21 15:54:32 *PST* 2013 by *lamport*

Test 1: 5248 distinct initial states 151056 full initial states

$$
\begin{aligned}
IInit \triangleq\ & \wedge num \in [Procs \to Nat] \\
& \wedge flag \in [Procs \to \text{BOOLEAN}] \\
& \wedge unchecked \in [Procs \to \text{SUBSET } Procs] \\
& \wedge max \in [Procs \to \{0\}] \setminus^* Nat] \\
& \wedge nxt \in [Procs \to \{1\}] \\
& \wedge pc \in [Procs \to \{\text{“ncs”}, \text{“e1”}, \text{“e2”}, \text{“e3”}, \text{“e4”}, \text{“w1”}, \text{“w2”}, \text{“cs”}\}] \\
& \wedge II
\end{aligned}
$$