MODULE $4state$

EXTENDS $Integers$, $FiniteSets$
CONSTANT $N$
ASSUME $N \in Nat \setminus \{0\}$
$Procs \triangleq 1 \,..\, N - 1$

Dijkstra's stabilizing 4 state token ring with processes

```
--algorithm TokenRing{
  variable c = [k ∈ 0 .. N ↦ (k%2)], up = [k ∈ 0 .. N ↦ IF k = N THEN FALSE ELSE TRUE];
    variable c = [k ∈ 0 .. N ↦ 0], up = [k ∈ 0 .. N ↦ IF k = 0 THEN TRUE ELSE FALSE];

   fair process ( j ∈ Procs )

    { J0: while ( TRUE )
        { either
             { await c[self] ≠ c[(self − 1)];
                  c[self] := c[(self − 1)];
                  up[self] := TRUE;
                 }
          or
          { await c[self] = c[(self + 1)] ∧ up[self] = TRUE ∧ up[(self + 1)] = FALSE;
                up[self] := FALSE;
               }
           }
     }
   fair process ( i ∈ {0} )
    { I0: while ( TRUE )
           { await (c[self] = c[1] ∧ up[1] = FALSE);
              c[self] := (c[self] + 1)%2;
            }
     }
    fair process ( k ∈ {N} )

    { N0: while ( TRUE )
          up[self] := FALSE;                 \* It is wrong to assign 'up' value here, because what if program executes process in pro
            { await c[self] ≠ c[(self − 1)];
               c[self] := c[(self − 1)];
             }
     }
 }
```

************************************************************

BEGIN TRANSLATION
VARIABLES $c$, $up$

$vars \triangleq \langle c, up \rangle$

$ProcSet \triangleq (Procs) \cup (\{0\}) \cup (\{N\})$

$Init \triangleq$ Global variables
$\land c = [k \in 0 \,..\, N \mapsto 0]$
$\land up = [k \in 0 \,..\, N \mapsto \text{IF } k = 0 \text{ THEN TRUE ELSE FALSE}]$

$$j(self) \triangleq \lor \land c[self] \neq c[(self - 1)]$$
$$\land c' = [c \text{ EXCEPT } ![self] = c[(self - 1)]]$$
$$\land up' = [up \text{ EXCEPT } ![self] = \text{TRUE}]$$
$$\lor \land c[self] = c[(self + 1)] \land up[self] = \text{TRUE} \land up[(self + 1)] = \text{FALSE}$$
$$\land up' = [up \text{ EXCEPT } ![self] = \text{FALSE}]$$
$$\land c' = c$$

$$i(self) \triangleq \land (c[self] = c[1] \land up[1] = \text{FALSE})$$
$$\land c' = [c \text{ EXCEPT } ![self] = (c[self] + 1)\%2]$$
$$\land up' = up$$

$$k(self) \triangleq \land c[self] \neq c[(self - 1)]$$
$$\land c' = [c \text{ EXCEPT } ![self] = c[(self - 1)]]$$
$$\land up' = up$$

$$Next \triangleq (\exists\, self \in Procs : j(self))$$
$$\lor (\exists\, self \in \{0\} : i(self))$$
$$\lor (\exists\, self \in \{N\} : k(self))$$

$$Spec \triangleq \land Init \land \Box[Next]_{vars}$$
$$\land \forall\, self \in Procs : \text{WF}_{vars}(j(self))$$
$$\land \forall\, self \in \{0\} : \text{WF}_{vars}(i(self))$$
$$\land \forall\, self \in \{N\} : \text{WF}_{vars}(k(self))$$

END TRANSLATION

$$Tokens \triangleq Cardinality(\{x \in Procs : c[x] \neq c[(x - 1)] \lor (c[x] = c[(x + 1)] \land up[x] = \text{TRUE} \land up[(x + 1)] = \text{FALS}$$
$$+ \text{IF } (c[0] = c[1]) \land up[1] = \text{FALSE THEN } 1 \text{ ELSE } 0$$
$$+ \text{IF } c[N] \neq c[(N - 1)] \text{ THEN } 1 \text{ ELSE } 0$$
$InvProp \triangleq Tokens = 1$
$Stabilization \triangleq \Diamond InvProp$
$LowerBound \triangleq Tokens \geq 1$
$NotIncrease \triangleq \Box[Tokens' \leq Tokens]_{vars}$
$Decrease \triangleq \forall\, m \in 1 \,..\, N + 1 : \Box\Diamond(Tokens \leq m)$
$TypeOK \triangleq \forall\, x \in 0 \,..\, N : c[x] < 2$

Dijkstra's stabilizing 4 State token ring algorithm. Made by
Akshay $Kumar - 50169103$
Rohin $Mittal - 50168799$