

MODULE *LockMutex*

This module defines *LM* to be the abstract program defined in Figure 4.6 of the book “A Science of Concurrent Programs” by *Leslie Lamport*, when the semaphores are assumed to be weakly fair. The module is imported by module *OneBit* that describes the book’s One Bit algorithm.

EXTENDS *Naturals*

This is a trivial two-process mutual exclusion algorithm using a weakly fair binary semaphore, aka a lock.

VARIABLES *sem*, *pc*

$v \triangleq \langle sem, pc \rangle$

$TypeOK \triangleq \wedge pc \in [\{0, 1\} \rightarrow \{\text{“ncs”}, \text{“wait”}, \text{“cs”}, \text{“exit”}\}]$
 $\wedge sem \in \{0, 1\}$

$Init \triangleq \wedge sem = 1$
 $\wedge pc = [p \in \{0, 1\} \mapsto \text{“ncs”}]$

$Ncs(p) \triangleq \wedge pc[p] = \text{“ncs”}$
 $\wedge \text{TRUE}$
 $\wedge pc' = [pc \text{ EXCEPT } ![p] = \text{“wait”}]$
 $\wedge sem' = sem$

$Wait(p) \triangleq \wedge pc[p] = \text{“wait”}$
 $\wedge sem = 1$
 $\wedge sem' = 0$
 $\wedge pc' = [pc \text{ EXCEPT } ![p] = \text{“cs”}]$

$Cs(p) \triangleq \wedge pc[p] = \text{“cs”}$
 $\wedge \text{TRUE}$
 $\wedge pc' = [pc \text{ EXCEPT } ![p] = \text{“exit”}]$
 $\wedge sem' = sem$

$Exit(p) \triangleq \wedge pc[p] = \text{“exit”}$
 $\wedge sem' = 1$
 $\wedge pc' = [pc \text{ EXCEPT } ![p] = \text{“ncs”}]$

$PSpec(p) \triangleq Ncs(p) \vee Wait(p) \vee Cs(p) \vee Exit(p)$

$Next \triangleq (\exists p \in \{0, 1\} : PSpec(p))$

The fairness property assumed of *LM* is weak fairness of each process action except the *Ncs* action, which the process executes to try to enter the critical section. Theorem 4.8 of the book implies that this fairness property for each process *p* can be written as the single *WF* property in the formula.

$LM \triangleq \wedge Init \wedge \Box[Next]_v$
 $\wedge \forall p \in \{0, 1\} : WF_v((pc[p] \neq \text{“ncs”}) \wedge PSpec(p))$

\ * Modification History

\ * Last modified Sun Oct 20 16:33:16 CEST 2024 by lamport

* Created *Thu Feb 09 10:20:42 PST 2023* by *lamport*