

Allen B. Downey

# The Little Book of Semaphores

Version 2.2.1

1

http://www.greenteapress.com/semaphores/LittleBookOfSemaphores.pdf

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

2

#### TLBOS, Chapter 3

## **Basic synchronization patterns**

#### 3.1 Signaling

Signaling makes it possible to guarantee that a section of code in one thread will run before a section of code in another thread; in other words. it solves the **serialization problem**.

The semaphore in the next program guarantee that the process A has completed the assignment to the variable x before the process B begins its assignment to the same variable.

## 3.1 Signaling (3.1.signaling.pml)

```
#define wait(sem)
                       atomic { sem > 0; sem-- }
16
   #define signal(sem) sem++
17
18
   byte sem = 0
19
  byte x = 0
20
21 proctype A() {
22
     x = 1
23
     signal(sem)
24 }
25
26
   proctype B() {
27
     wait(sem)
28
     x = 2
29 }
30
31 init {
32
     atomic { run A(); run B() }
33
      _nr_pr == 1
34
     assert( x == 2 )
35 }
```

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

### 3.1 Signaling (3.1.signaling.pml)

```
$ spin -run 3.1.signaling.pml
(Spin Version 6.4.8 -- 2 March 2018)
        + Partial Order Reduction
Full statespace search for:
        never claim
                               - (none specified)
        assertion violations
        cycle checks
                               - (disabled by -DSAFETY)
        invalid end states
State-vector 28 byte, depth reached 11, errors: 0
```

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

5

7

TLBOS, Chapter 3

## **Basic synchronization patterns**

#### 3.3 Rendezvous

The idea is that two threads rendezvous at a point of execution, and neither is allowed to proceed until both have arrived.

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

6



Claude Lelouch, 1976, 8 min 38 seconds



VK, 2018 - The Little Book of Semaphores

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

INF646 Métodos Formales

```
3.3 Rendezvous (3.3.0.rendezvous.pml)
                                                                                    3.3 Rendezvous (3.3.0.rendezvous.pml)
$ cat -n 3.3.0.rendezvous.pml
     1 /* The Little Book of Semaphores (2.2.1)
                                                                                     20 int x = 0
           by A. Downey
                                                                                     22 proctype A() {
     3
     4
           Chapter 3. Basic synchronization patterns
                                                                                     x = 10*x + 1
                                                                                     24 \times = 10*x + 2
     6
           3.1 Signaling
                                                                                     25 }
     7
           3.3 Rendezvous
                                                                                     26
     8
                                                   Thread 2
                          Thread A
                                                                                     27 proctype B() {
     9
                          1 statement a1
                                                   1 statement b1
                                                                                     28 \quad x = 10*x + 3
    10
                          2 statement a2
                                                   2 statement b2
                                                                                     x = 10*x + 4
    11
                                                                                     30 }
    12
           We want to guarantee that a1 happens before b2 and b1 happens
                                                                                     31
before a2:
                                                                                     32 init {
    13
               a1,b1,b2,a2; a1,b1,a2,b2; b1,a1,a2,b2; b1,a1,b2,a2
                                                                                          atomic { run A(); run B() }
                                                                                          nr pr == 1
    14
           prohibiting
                                                                                          printf("x = %d\n", x)
    15
               b1,b2,a1,a2; a1,a2,b1,b2
                                                                                          assert(x==1234 || x==1324 || x==1342 || x==3412 || x==3142 ||
    16
                                                                                     36
    17
           3.3.0.rendezvous.pml: all 6 possible sequences
                                                                                 x = 3124)
    18 */
                                                                                     37 /* must be prohibited: 3412 and 1234 */
    19
                                                                                     38 }
. . .
  INF646 Métodos Formales
                                                                           9
                                                                                    INF646 Métodos Formales
                                                                                                                                                            10
                            VK, 2018 - The Little Book of Semaphores
                                                                                                              VK, 2018 - The Little Book of Semaphores
  3.3 Rendezvous (3.3.0.rendezvous.pml)
                                                                                    3.3 Rendezvous (3.3.0.rendezvous.pml)
$ spin 3.3.0.rendezvous.pml
      x = 1342
                                                                                 $ spin -run 3.3.0.rendezvous.pml
3 processes created
                                                                                 (Spin Version 6.4.8 -- 2 March 2018)
$ spin 3.3.0.rendezvous.pml
                                                                                         + Partial Order Reduction
      x = 1234
                                                                                 Full statespace search for:
3 processes created
                                                                                         never claim
                                                                                                                   (none specified)
                                                                                         assertion violations
$ spin 3.3.0.rendezvous.pml
                                                                                         cycle checks
      x = 1234
                                                                                                                  - (disabled by -DSAFETY)
3 processes created
                                                                                         invalid end states
$ spin 3.3.0.rendezvous.pml
                                                                                 State-vector 36 byte, depth reached 12, errors: 0
      x = 3142
3 processes created
$ spin 3.3.0.rendezvous.pml
                                                                                 Only these 6 sequences are possible, but two of them are prohibited!
      x = 3412
3 processes created
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                          11
                                                                                    INF646 Métodos Formales
                                                                                                              VK, 2018 - The Little Book of Semaphores
                                                                                                                                                            12
```

```
3.3.2 Rendezvous (efficient) solution (3.3.2a.rendezvous.pml)
                                                                                     3.3.2 Rendezvous (efficient) solution (3.3.2a.rendezvous.pml)
$ cat -n 3.3.2a.rendezvous.pml
    1 /* The Little Book of Semaphores (2.2.1)
                                                                                      20 #define wait(sem) atomic { sem > 0; sem-- }
                                                                                      21 #define signal(sem) sem++
           by A. Downey
     4
           Chapter 3. Basic synchronization patterns
                                                                                      23 byte aArrived = 0, bArrived = 0
     5
                                                                                      24 \text{ int } x = 0
     6
           3.1 Signaling
                                                                                      25
     7
           3.3 Rendezvous
                                                                                      26 proctype A() {
                                                                                          x = 10*x + 1
     8
                           Thread A
                                                    Thread 2
     9
                           1 statement a1
                                                    1 statement b1
                                                                                           signal(aArrived)
                                                                                                               # a) "llegaré en 10 minutos"
                           2 statement a2
                                                    2 statement b2
                                                                                      29
                                                                                           wait(bArrived)
                                                                                                               # b) llega en 8, debe esperar (context switch)
    10
    11
                                                                                      30
                                                                                           x = 10*x + 2
    12
                                                                                      31 }
           We want to guarantee that all happens before b2 and b1 happens
before a2:
                                                                                      32
    13
               a1,b1,b2,a2; a1,b1,a2,b2; b1,a1,a2,b2; b1,a1,b2,a2
                                                                                      33 proctype B() {
                                                                                           x = 10*x + 3
    14
           prohibiting
               b1,b2,a1,a2; a1,a2,b1,b2
    15
                                                                                           signal(bArrived)
                                                                                                               # c) "ya llegué"
    16
                                                                                           wait(aArrived)
                                                                                                               # d) puede seguir sin cambio del contexto (1342)
                                                                                           x = 10*x + 4
                                                                                      37
    17
           3.3.2a Rendezvous solution (efficient)
    18 */
                                                                                      38 }
    19
                                                                                      39
. . .
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                           13
                                                                                     INF646 Métodos Formales
                                                                                                                                                             14
                                                                                                               VK, 2018 - The Little Book of Semaphores
  3.3.2 Rendezvous (efficient) solution (3.3.2a.rendezvous.pml)
                                                                                     3.3.2 Rendezvous (efficient) solution (3.3.2a.rendezvous.pml)
                                                                                  $ spin -run 3.3.2a.rendezvous.pml
    40 init {
       atomic { run A(); run B() }
                                                                                  (Spin Version 6.4.8 -- 2 March 2018)
        _nr_pr == 1
                                                                                          + Partial Order Reduction
    43 assert(x!=1234 && x!=3412)
                                                                                  Full statespace search for:
    44 }
                                                                                          never claim
                                                                                                                    - (none specified)
                                                                                          assertion violations
                                                                                          cycle checks
                                                                                                                    - (disabled by -DSAFETY)
                                                                                          invalid end states
                                                                                  State-vector 36 byte, depth reached 15, errors: 0
                                                                                  . . .
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                           15
                                                                                     INF646 Métodos Formales
                                                                                                               VK, 2018 - The Little Book of Semaphores
                                                                                                                                                             16
```

```
3.3.2 Rendezvous (less efficient) solution (3.3.2b.rendezvous.pml)
                                                                                      3.3.2 Rendezvous (less efficient) solution (3.2.2b.rendezvous.pml)
$ cat -n 3.3.2b.rendezvous.pml
     1 /* The Little Book of Semaphores (2.2.1)
                                                                                        20 #define wait(sem) atomic { sem > 0; sem-- }
                                                                                       21 #define signal(sem) sem++
           by A. Downey
     3
     4
           Chapter 3. Basic synchronization patterns
                                                                                       23 byte aArrived = 0, bArrived = 0
     5
                                                                                        24 \text{ int } x = 0
     6
           3.1 Signaling
                                                                                       25
     7
           3.3 Rendezvous
                                                                                       26 proctype A() {
                                                    Thread 2
                                                                                       27 \quad x = 10*x + 1
     8
                           Thread A
     9
                           1 statement a1
                                                    1 statement b1
                                                                                             wait(bArrived)
                                                                                                                  # a) "¿cuándo llegaras?" (context switch)
                           2 statement a2
                                                    2 statement b2
                                                                                       29
                                                                                             signal(aArrived)
                                                                                                                  # d) "pardon, ya estoy" (1324 o 1342)
    10
    11
                                                                                        30
                                                                                             x = 10*x + 2
    12
           We want to guarantee that all happens before b2 and b1 happens
                                                                                        31 }
before a2:
                                                                                        32
    13
                a1,b1,b2,a2; a1,b1,a2,b2; b1,a1,a2,b2; b1,a1,b2,a2
                                                                                        33 proctype B() {
                                                                                            x = 10*x + 3
    14
           prohibiting
               b1,b2,a1,a2; a1,a2,b1,b2
    15
                                                                                             signal(bArrived)
                                                                                                                  # b) "ya llegué"
                                                                                                                  # c) "pero tú no estás" (context switch extra)
    16
                                                                                             wait(aArrived)
                                                                                             x = 10*x + 4
    17
           3.3.2b Rendezvous solution (less efficient)
                                                                                       37
    18 */
                                                                                        38 }
    19
                                                                                       39
. . .
  INF646 Métodos Formales
                             VK, 2018 - The Little Book of Semaphores
                                                                            17
                                                                                      INF646 Métodos Formales
                                                                                                                                                                18
                                                                                                                 VK, 2018 - The Little Book of Semaphores
  3.3.2 Rendezvous (less efficient) solution (3.3.2b.rendezvous.pml)
                                                                                      3.3.2 Rendezvous (less efficient) solution (3.3.2b.rendezvous.pml)
                                                                                   $ spin -a 3.3.2b.rendezvous.pml
    40 init {
         atomic { run A(); run B() }
                                                                                   (Spin Version 6.4.8 -- 2 March 2018)
         nr pr == 1
                                                                                            + Partial Order Reduction
       assert(x!=1234 && x!=3412)
    43
                                                                                   Full statespace search for:
    44 }
    45
                                                                                            never claim
                                                                                                                      - (none specified)
                                                                                            assertion violations
                                                                                            cycle checks
                                                                                                                     - (disabled by -DSAFETY)
                                                                                            invalid end states
                                                                                   State-vector 36 byte, depth reached 15, errors: 0
                                                                                   . . .
  INF646 Métodos Formales
                             VK, 2018 - The Little Book of Semaphores
                                                                            19
                                                                                      INF646 Métodos Formales
                                                                                                                 VK, 2018 - The Little Book of Semaphores
                                                                                                                                                               20
```

```
3.3.3 Rendezvous Deadlock #1 (3.3.3.rendezvous.pml)
                                                                                     3.3.3 Rendezvous Deadlock #1 (3.3.3.rendezvous.pml)
$ cat -n 3.2.3.rendezvous.pml
     1 /* The Little Book of Semaphores (2.1.5)
                                                                                       20 #define wait(sem) atomic { sem > 0; sem-- }
                                                                                       21 #define signal(sem) sem++
           by A. Downey
     4
           Chapter 3. Basic synchronization patterns
                                                                                       23 byte aArrived = 0, bArrived = 0;
     5
                                                                                       24 \text{ int } x = 0
     6
           3.1 Signaling
                                                                                       25
     7
           3.2 Rendezvous
                                                                                       26 proctype A() {
     8
                           Thread A
                                                    Thread 2
                                                                                       x = 10*x + 1
     9
                           1 statement a1
                                                    1 statement b1
                                                                                            wait(bArrived)
                           2 statement a2
                                                    2 statement b2
                                                                                       29
                                                                                            signal(aArrived)
    10
    11
                                                                                            x = 10*x + 2
    12
           We want to guarantee that all happens before b2 and b1 happens
                                                                                       31 }
before a2:
                                                                                       32
    13
               a1,b1,b2,a2; a1,b1,a2,b2; b1,a1,a2,b2; b1,a1,b2,a2
                                                                                       33 proctype B() {
                                                                                       34 \quad x = 10*x + 3
    14
           prohibiting
    15
               b1,b2,a1,a2; a1,a2,b1,b2
                                                                                            wait(aArrived)
    16
                                                                                            signal(bArrived)
                                                                                           x = 10*x + 4
           3.2.3 Deadlock #1
    17
                                                                                       37
    18 */
                                                                                       38 }
    19
                                                                                       39
. . .
  INF646 Métodos Formales
                             VK, 2018 - The Little Book of Semaphores
                                                                           21
                                                                                     INF646 Métodos Formales
                                                                                                                                                              22
                                                                                                                VK, 2018 - The Little Book of Semaphores
  3.3.3 Rendezvous Deadlock #1 (3.3.3.rendezvous.pml)
                                                                                     3.3.3 Rendezvous Deadlock #1 (3.3.3.rendezvous.pml)
                                                                                  $ spin -run 3.3.3.rendezvous.pml
                                                                                  pan:1: invalid end state (at depth 3)
    40 init {
    41 atomic { run A(); run B() }
                                                                                  pan: wrote 3.3.3.rendezvous.pml.trail
    42 nr pr == 1
    43 assert(x!=1234 && x!=3412)
                                                                                  (Spin Version 6.4.8 -- 2 March 2018)
    44 }
                                                                                  Warning: Search not completed
                                                                                           + Partial Order Reduction
                                                                                  Full statespace search for:
                                                                                           never claim
                                                                                                                    - (none specified)
                                                                                           assertion violations
                                                                                           cycle checks
                                                                                                                    (disabled by -DSAFETY)
                                                                                           invalid end states
                                                                                  State-vector 36 byte, depth reached 4, errors: 1
                                                                                  . . .
  INF646 Métodos Formales
                             VK, 2018 - The Little Book of Semaphores
                                                                           23
                                                                                     INF646 Métodos Formales
                                                                                                                VK, 2018 - The Little Book of Semaphores
                                                                                                                                                              24
```

```
3.3.3 Rendezvous Deadlock #1 (3.3.3.rendezvous.pml)
                                                                                    3.3.3 Rendezvous Deadlock #1 (3.3.3.rendezvous.pml)
$ spin 3.3.3.rendezvous.pml
                                                                                 $ spin -p 3.3.3.rendezvous.pml
                                                                                          proc - (:root:) creates proc 0 (:init:)
      timeout
                                                                                 Starting A with pid 1
#processes: 3
                aArrived = 0
                                                                                          proc 0 (:init::1) creates proc 1 (A)
                bArrived = 0
                                                                                          proc 0 (:init::1) 3.3.3.rendezvous.pml:41 (state 1)
                                                                                                                                                   [(run A())]
                x = 31
                                                                                 Starting B with pid 2
                                                                                          proc 0 (:init::1) creates proc 2 (B)
        proc 2 (B:1) 3.3.3.rendezvous.pml:35 (state 4)
                                                                                   2:
 4:
        proc 1 (A:1) 3.3.3.rendezvous.pml:28 (state 4)
                                                                                          proc 0 (:init::1) 3.3.3.rendezvous.pml:41 (state 2)
 4:
                                                                                                                                                   [(run B())]
        proc 0 (:init::1) 3.3.3.rendezvous.pml:42 (state 4)
                                                                                          proc 1 (A:1) 3.3.3.rendezvous.pml:27 (state 1) [x = ((10*x)+1)]
                                                                                          proc 2 (B:1) 3.3.3.rendezvous.pml:34 (state 1) [x = ((10*x)+3)]
3 processes created
                                                                                        timeout
                                                                                 #processes: 3
                                                                                                  aArrived = 0
                                                                                                  bArrived = 0
                                                                                                  x = 13
                                                                                          proc 2 (B:1) 3.3.3.rendezvous.pml:35 (state 4)
                                                                                          proc 1 (A:1) 3.3.3.rendezvous.pml:28 (state 4)
                                                                                          proc 0 (:init::1) 3.3.3.rendezvous.pml:42 (state 4)
                                                                                 3 processes created
  INF646 Métodos Formales
                                                                          25
                                                                                    INF646 Métodos Formales
                                                                                                                                                            26
                            VK, 2018 - The Little Book of Semaphores
                                                                                                              VK, 2018 - The Little Book of Semaphores
  3.4.0 Shared variable (3.4.0.shared_var.pml)
                                                                                    3.4.0 Shared variable (3.4.0.shared_var.pml)
$ cat -n 3.4.0.shared_var.pml
     1 /* The Little Book of Semaphores (2.2.1)
                                                                                      13 byte count = 0
           by A. Downey
                                                                                      14
     3
                                                                                      15 proctype Th(byte i) {
     4
           Chapter 3. Basic synchronization patterns
                                                                                          byte temp
     5
                                                                                     17
     6
           3.4 Mutex
                                                                                     18
                                                                                          temp = count
     7
                          Thread A
                                                      Thread B
                                                                                           count = temp + 1
                                                                                      19
     8
                          1 count = count + 1
                                                      1 count = count + 1
                                                                                      20
                                                                                          printf("%c: count=%d\n",i,count)
     9
                                                                                      21 }
    10
           3.4.0.shared_var.pml
                                                                                      22
    11 */
                                                                                      23 init {
                                                                                          atomic { run Th('A'); run Th('B') }
    12
                                                                                          nr pr == 1
                                                                                      26 assert(count==2)
                                                                                      27 }
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                          27
                                                                                    INF646 Métodos Formales
                                                                                                              VK, 2018 - The Little Book of Semaphores
                                                                                                                                                            28
```

```
3.4.0 Shared variable (3.4.0.shared_var.pml)
                                                                                   3.4.2 Mutex (3.4.2.mutex.pml)
                                                                                 $ cat -n 3.4.2.mutex.pml
$ spin 3.4.0.shared_var.pml
              B: count=1
                                                                                      1 /* The Little Book of Semaphores (2.2.1)
          A: count=2
                                                                                            by A. Downey
3 processes created
                                                                                      3
                                                                                            Chapter 3. Basic synchronization patterns
                                                                                            3.4 Mutex
                                                                                                       Thread A
                                                                                                                                   Thread B
$ spin 3.4.0.shared_var.pml
              B: count=1
                                                                                                       1 mutex.wait()
                                                                                                                                   1 mutex.wait()
          A: count=1
                                                                                      9
                                                                                                       2 # critical section
                                                                                                                                   2 # critical section
spin: 3.4.0.shared_var.pml:26, Error: assertion violated
                                                                                     10
                                                                                                       3 count = count + 1
                                                                                                                                   3 count = count + 1
spin: text of failed assertion: assert((count==2))
                                                                                                       4 mutex.signal()
                                                                                                                                   4 mutex.signal()
                                                                                     11
#processes: 1
                                                                                     12
                                                                                     13
                count = 1
                                                                                            3.4.2.mutex.pml
       proc 0 (:init::1) 3.4.0.shared_var.pml:26 (state 5)
                                                                                     14 */
3 processes created
                                                                                     15
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                          29
                                                                                   INF646 Métodos Formales
                                                                                                             VK, 2018 - The Little Book of Semaphores
                                                                                                                                                           30
  3.4.2 Mutex (3.4.2.mutex.pml)
                                                                                   3.4.2 Mutex (3.4.2.mutex.pml)
                                                                                 $ spin -run 3.4.2.mutex.pml
   16 #define wait(sem) atomic { sem > 0; sem-- }
   17 #define signal(sem) sem++
                                                                                 (Spin Version 6.4.8 -- 2 March 2018)
                                                                                         + Partial Order Reduction
   18
    19 byte mutex = 1
    20 byte count = 0
                                                                                 Full statespace search for:
                                                                                         never claim
                                                                                                                  - (none specified)
    22 proctype Th(byte i) {
                                                                                         assertion violations
        byte temp
                                                                                         cycle checks
                                                                                                                  - (disabled by -DSAFETY)
   24
                                                                                         invalid end states
   25 wait(mutex)
   26
          temp = count
                                                                                 State-vector 28 byte, depth reached 15, errors: 0
   27
           count = temp + 1
    28 signal(mutex)
                                                                                 . . .
    29 }
    30
    31 init {
   32 atomic { run Th('A'); run Th('B') }
    33
        nr pr == 1
    34 assert(count==2)
    35 }
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                          31
                                                                                   INF646 Métodos Formales
                                                                                                             VK, 2018 - The Little Book of Semaphores
                                                                                                                                                           32
```

```
3.5.1 Multiplex (3.5.1.multiplex.pml)
                                                                                     3.5.1 Multiplex (3.5.1.multiplex.pml)
                                                                                       15 #define wait(sem) atomic { sem > 0; sem-- }
$ cat -n 3.5.1.multiplex.pml | expand
                                                                                           #define signal(sem) sem++
                                                                                       17
        /* The Little Book of Semaphores (2.2.1)
                                                                                       18
                                                                                           #define LIMIT 3
            by A. Downey
                                                                                       19
     3
                                                                                           byte multiplex=LIMIT, cs=0
     4
            Chapter 3. Basic synchronization patterns
                                                                                       21
     5
                                                                                           proctype Th(byte i) {
            3.4 Multiplex
                                                                                             wait(multiplex)
     7
                        Thread i
                                                                                       24
                                                                                               cs++ /* atomic inc by Promela */
     8
                        1 multiplex.wait()
                                                                                       25
                                                                                               assert(cs <= LIMIT)</pre>
     9
                            # critical section
                                                                                               cs-- /* atomic dec by Promela */
                                                                                       26
    10
                        3 multiplex.signal()
                                                                                             signal(multiplex)
                                                                                       27
    11
                                                                                       28
    12
            3.5.1.multiplex.pml
                                                                                       29
    13 */
                                                                                       30
                                                                                           init {
    14
                                                                                             byte i
                                                                                       32
                                                                                       33
                                                                                             atomic {
                                                                                               for (i : 1 .. 9) {
                                                                                       35
                                                                                                 run Th(i)
                                                                                       36
                                                                                       37
                                                                                       38
  INF646 Métodos Formales
                                                                           33
                                                                                     INF646 Métodos Formales
                                                                                                                VK, 2018 - The Little Book of Semaphores
                                                                                                                                                              34
                             VK, 2018 - The Little Book of Semaphores
  3.5.1 Multiplex (3.5.1.multiplex.pml)
                                                                                     3.6.2 Barrier non-solution (3.6.2a.barrier_nonsol.pml)
$ spin -run 3.5.1.multiplex.pml | expand
                                                                                   $ cat -n 3.6.2a.barrier_nonsol.pml | expand
(Spin Version 6.4.8 -- 2 March 2018)
        + Partial Order Reduction
                                                                                        1 /* The Little Book of Semaphores (2.2.1)
                                                                                               by A. Downey
Full statespace search for:
                                                                                        3
        never claim
                                 - (none specified)
                                                                                               Chapter 3. Basic synchronization patterns
        assertion violations
        cycle checks
                                 - (disabled by -DSAFETY)
                                                                                               3.6 Barrier
        invalid end states
                                                                                               3.6.2 Barrier non-solution
State-vector 84 byte, depth reached 86, errors: 0
                                                                                               vk, 2017
                                                                                       10
                                                                                       11
                                                                                           #define THREADS 10
                                                                                                                /* value for threads number */
                                                                                                                 /* value for barrier limit */
unreached in proctype Th
                                                                                       13 #define N
        (0 of 8 states)
                                                                                       14
unreached in init
                                                                                       15 #define wait(sem)
                                                                                                                atomic { sem > 0; sem-- }
                                                                                       16 #define signal(sem) sem++
        (0 of 11 states)
                                                                                       17
pan: elapsed time 0.35 seconds
pan: rate 1372128.6 states/second
  INF646 Métodos Formales
                             VK, 2018 - The Little Book of Semaphores
                                                                           35
                                                                                     INF646 Métodos Formales
                                                                                                                VK, 2018 - The Little Book of Semaphores
                                                                                                                                                              36
```

```
3.6.2 Barrier non-solution (3.6.2a.barrier_nonsol.pml)
  3.6.2 Barrier non-solution (3.6.2a.barrier_nonsol.pml)
    18
       byte count=0, mutex=1, barrier=0
                                                                                      39
                                                                                          init {
    19
                                                                                       40
                                                                                               byte i
        proctype Th(byte i) {
                                                                                      41
                                                                                      42
    21
            byte temp
                                                                                               atomic {
    22
                                                                                      43
                                                                                                   for (i: 1 .. THREADS) {
    23
                                                                                      44
                                                                                                       run Th(i)
            do
    24
                                                                                      45
            :: wait(mutex)
    25
                     temp=count
                                                                                      46
    26
                     count=temp+1
                                                                                       47
    27
                signal(mutex)
    28
                if
    29
                 :: count == N ->
    30
                                  signal(barrier)
    31
                :: else
    32
    33
                wait(barrier)
    34
                printf("Th(%d): count = %d\n",i,count)
    35
                break
    36
            od
    37 }
    38
  INF646 Métodos Formales
                                                                           37
                                                                                                                                                             38
                            VK, 2018 - The Little Book of Semaphores
                                                                                     INF646 Métodos Formales
                                                                                                               VK, 2018 - The Little Book of Semaphores
  3.6.2 Barrier non-solution (3.6.2a.barrier_nonsol.pml)
                                                                                     3.6.2 Barrier non-solution (3.6.2a.barrier_nonsol.pml)
$ spin 3.6.2a.barrier_nonsol.pml | expand
                                                                                  $ spin 3.6.2a.barrier_nonsol.pml | expand
                                       Th(8): count = 5
                                                                                             Th(1): count = 7
      timeout
                                                                                        timeout
                                                                                  #processes: 11
#processes: 11
                count = 10
                                                                                                   count = 10
                mutex = 1
                                                                                                   mutex = 1
                barrier = 0
                                                                                                   barrier = 0
                                                                                           proc 10 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
109:
        proc 10 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
                                                                                  109:
109:
        proc 9 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                  109:
                                                                                           proc 9 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                          proc 8 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
109:
        proc 8 (Th:1) 3.6.2a.barrier nonsol.pml:37 (state 20) <valid end
                                                                                  109:
                                                                                  109:
                                                                                           proc 7 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
state>
109:
        proc 7 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                  109:
                                                                                           proc 6 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
109:
        proc 6 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                  109:
                                                                                           proc 5 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
        proc 5 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
                                                                                  109:
                                                                                           proc 4 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
109:
109:
        proc 4 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                  109:
                                                                                           proc 3 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
109:
        proc 3 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                  109:
                                                                                           proc 2 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
109:
        proc 2 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
                                                                                  109:
                                                                                           proc 1 (Th:1) 3.6.2a.barrier nonsol.pml:37 (state 20) <valid end
109:
        proc 1 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                  state>
        proc 0 (:init::1) 3.6.2a.barrier nonsol.pml:47 (state 11) <valid end
                                                                                  109:
                                                                                           proc 0 (:init::1) 3.6.2a.barrier nonsol.pml:47 (state 11) <valid end
109:
state>
                                                                                  state>
                                                                                  11 processes created
11 processes created
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                           39
                                                                                     INF646 Métodos Formales
                                                                                                               VK, 2018 - The Little Book of Semaphores
```

```
3.6.2 Barrier non-solution (3.6.2a.barrier_nonsol.pml)
                                                                                   3.6.2 Barrier non-solution (3.6.2a.barrier_nonsol.pml)
$ spin 3.6.2a.barrier nonsol.pml | expand
                                                                                $ spin 3.6.2a.barrier nonsol.pml | expand
                                                                                                                       Th(8): count = 5
      timeout
                                                                                                           Th(5): count = 5
#processes: 11
                                                                                      timeout
                                                                                #processes: 11
                count = 10
                mutex = 1
                                                                                                count = 10
                barrier = 0
                                                                                                mutex = 1
                                                                                                barrier = 0
104:
        proc 10 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
        proc 9 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
104:
                                                                                114:
                                                                                        proc 10 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
        proc 8 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                        proc 9 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
104:
                                                                                114:
        proc 7 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                114:
                                                                                         proc 8 (Th:1) 3.6.2a.barrier nonsol.pml:37 (state 20) <valid end
104:
104:
        proc 6 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
                                                                                state>
        proc 5 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
104:
                                                                                114:
                                                                                        proc 7 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
104:
        proc 4 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                114:
                                                                                        proc 6 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
        proc 3 (Th:1) 3.6.2a.barrier_nonsol.pml:33 (state 14)
                                                                                        proc 5 (Th:1) 3.6.2a.barrier_nonsol.pml:37 (state 20) <valid end
104:
                                                                                114:
        proc 2 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
104:
                                                                                state>
104:
        proc 1 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                114:
                                                                                        proc 4 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
104:
        proc 0 (:init::1) 3.6.2a.barrier nonsol.pml:47 (state 11) <valid end
                                                                                114:
                                                                                        proc 3 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
state>
                                                                                114:
                                                                                        proc 2 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
                                                                                114:
                                                                                        proc 1 (Th:1) 3.6.2a.barrier nonsol.pml:33 (state 14)
11 processes created
                                                                                        proc 0 (:init::1) 3.6.2a.barrier_nonsol.pml:47 (state 11) <valid end
                                                                                114:
                                                                                state>
                                                                                11 processes created
  INF646 Métodos Formales
                                                                         41
                                                                                                                                                         42
                            VK, 2018 - The Little Book of Semaphores
                                                                                   INF646 Métodos Formales
                                                                                                             VK, 2018 - The Little Book of Semaphores
  3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                   3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                    18 byte count=0, mutex=1, barrier=0
                                                                                        bit bar[THREADS+1]
$ cat -n 3.6.3.barrier_nonsol.pml | expand
                                                                                    20
                                                                                        proctype Th(byte i) {
                                                                                    21
     1 /* The Little Book of Semaphores (2.2.1)
                                                                                    22
                                                                                            byte temp
            by A. Downey
                                                                                    23
     3
                                                                                    24
                                                                                            do
     4
            Chapter 3. Basic synchronization patterns
                                                                                    25
                                                                                             :: wait(mutex)
     5
                                                                                    26
                                                                                                    temp=count
     6
            3.6 Barrier
                                                                                    27
                                                                                                    count=temp+1
     7
            3.6.2 Barrier non-solution
                                                                                    28
                                                                                                signal(mutex)
     8
                                                                                    29
                                                                                                bar[i]=false
     9
            vk, 2017
                                                                                    30
    10
       */
                                                                                    31
                                                                                                :: count == N ->
    11
                                                                                    32
                                                                                                    bar[i]=true
       #define THREADS 5
                             /* value for threads number */
    12
                                                                                                    assert(!bar[1]||!bar[2]||!bar[3]||!bar[4]||!bar[5])
                                                                                    33
   13 #define N
                             /* value for barrier limit */
                                                                                    34
                                                                                                    signal(barrier)
   14
                                                                                    35
                                                                                                :: else
    15 #define wait(sem)
                            atomic { sem > 0; sem-- }
                                                                                    36
                                                                                                fi
    16 #define signal(sem) sem++
                                                                                    37
                                                                                                wait(barrier)
   17
                                                                                    38
                                                                                                printf("Th(%d): count = %d\n",i,count)
```

INF646 Métodos Formales VK, 2018 - The Little Book of Semaphores

INF646 Métodos Formales

od

break

39

40

41

43

VK, 2018 - The Little Book of Semaphores

```
3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                                3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                            $ spin -run (-E)3.6.3.barrier nonsol.pml | expand
. . .
    42
    43 init {
                                                                                            pan:1: assertion violated (((( !(bar[1])|| !(bar[2]))|| !(bar[3]))|| !
                                                                                            (bar[4]))|| !(bar[5])) (at depth 74)
    44
             byte i
                                                                                            pan: wrote 3.6.3.barrier_nonsol.pml.trail
    45
    46
             atomic {
    47
                  for (i: 1 .. THREADS) {
                                                                                            (Spin Version 6.4.8 -- 2 March 2018)
    48
                       run Th(i)
                                                                                            Warning: Search not completed
                                                                                                      + Partial Order Reduction
    49
    50
    51
                                                                                            Full statespace search for:
                                                                                                      never claim
                                                                                                                                   - (none specified)
                                                                                                      assertion violations
                                                                                                                                  - (disabled by -DSAFETY)
                                                                                                      cvcle checks
                                                                                                      invalid end states
                                                                                                                                  - (disabled by -E flag)
                                                                                            State-vector 64 byte, depth reached 74, errors: 1
                                                                                            . . .
                                                                                    45
                                                                                                                                                                                 46
  INF646 Métodos Formales
                                VK, 2018 - The Little Book of Semaphores
                                                                                                INF646 Métodos Formales
                                                                                                                             VK, 2018 - The Little Book of Semaphores
  3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                               3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
$ spin -t -p -q -l 3.6.3.barrier nonsol.pml | expand
using statement merging
                                                                                                    proc 4 (Th:1) 3.6.3.barrier nonsol.pml:25 (state 1)
                                                                                                                                                        [((mutex>0))]
      proc 0 (:init::1) 3.6.3.barrier_nonsol.pml:47 (state 1)
                                                                  [i = 1]
                                                                                                    proc 4 (Th:1) 3.6.3.barrier_nonsol.pml:25 (state 2)
                                                                                                                                                        [mutex = (mutex-1)]
               :init:(0):i = 1
       proc 0 (:init::1) 3.6.3.barrier nonsol.pml:47 (state 2)
                                                                  [((i<=5))]
                                                                                                    proc 4 (Th:1) 3.6.3.barrier nonsol.pml:26 (state 4)
                                                                                                                                                        [temp = count]
                                                                                             26:
Starting Th with pid 1
                                                                                                            Th(4):temp = 1
       proc 0 (:init::1) 3.6.3.barrier nonsol.pml:48 (state 3)
                                                                  [(run Th(i))]
                                                                                                    proc 4 (Th:1) 3.6.3.barrier nonsol.pml:27 (state 5)
                                                                                                                                                        [count = (temp+1)]
 3:
 4:
       proc 0 (:init::1) 3.6.3.barrier_nonsol.pml:47 (state 4)
                                                                  [i = (i+1)]
              :init:(0):i = 2
                                                                                             28:
                                                                                                    proc 4 (Th:1) 3.6.3.barrier_nonsol.pml:28 (state 6)
                                                                                                                                                        [mutex = (mutex+1)]
. . .
               :init:(0):i = 6
                                                                                                    proc 4 (Th:1) 3.6.3.barrier nonsol.pml:29 (state 7)
                                                                                                                                                        [bar[i] = 0]
       proc 0 (:init::1) 3.6.3.barrier_nonsol.pml:49 (state 5)
17:
                                                                   [else]
       proc 0 (:init::1) 3.6.3.barrier nonsol.pml:49 (state 6)
                                                                   [goto :b1]
       proc 0 (:init::1) 3.6.3.barrier_nonsol.pml:49 (state 9)
                                                                  [break]
                                                                                                    proc 3 (Th:1) 3.6.3.barrier_nonsol.pml:25 (state 1)
                                                                                                                                                        [((mutex>0))]
                                                                                                    proc 3 (Th:1) 3.6.3.barrier nonsol.pml:25 (state 2)
                                                                                                                                                        [mutex = (mutex-1)]
20:
       proc 5 (Th:1) 3.6.3.barrier_nonsol.pml:25 (state 1)
                                                           [((mutex>0))]
                                                                                                            mutex = 0
       proc 5 (Th:1) 3.6.3.barrier nonsol.pml:25 (state 2)
                                                           [mutex = (mutex-1)]
                                                                                                    proc 3 (Th:1) 3.6.3.barrier_nonsol.pml:26 (state 4)
20:
                                                                                             31:
                                                                                                                                                        [temp = count]
              mutex = 0
                                                                                                            Th(3):temp = 2
21:
       proc 5 (Th:1) 3.6.3.barrier_nonsol.pml:26 (state 4)
                                                           [temp = count]
                                                                                             32:
                                                                                                    proc 3 (Th:1) 3.6.3.barrier_nonsol.pml:27 (state 5)
                                                                                                                                                        [count = (temp+1)]
              Th(5):temp = 0
                                                                                                            count = 3
22:
       proc 5 (Th:1) 3.6.3.barrier nonsol.pml:27 (state 5)
                                                          [count = (temp+1)]
                                                                                             33:
                                                                                                    proc 3 (Th:1) 3.6.3.barrier nonsol.pml:28 (state 6)
                                                                                                                                                        [mutex = (mutex+1)]
       proc 5 (Th:1) 3.6.3.barrier_nonsol.pml:28 (state 6)
23:
                                                          [mutex = (mutex+1)]
                                                                                                    proc 3 (Th:1) 3.6.3.barrier nonsol.pml:29 (state 7)
                                                                                                                                                        [bar[i] = 0]
24:
       proc 5 (Th:1) 3.6.3.barrier_nonsol.pml:29 (state 7) [bar[i] = 0]
```

47

**INF646 Métodos Formales** 

VK, 2018 - The Little Book of Semaphores

**INF646 Métodos Formales** 

VK, 2018 - The Little Book of Semaphores

```
3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
35:
       proc 2 (Th:1) 3.6.3.barrier nonsol.pml:25 (state 1)
                                                              [((mutex>0))]
                                                                                                   43:
                                                                                                          proc 5 (Th:1) 3.6.3.barrier nonsol.pml:31 (state 8)
                                                                                                                                                                 [((count==5))]
       proc 2 (Th:1) 3.6.3.barrier_nonsol.pml:25 (state 2)
                                                              [mutex = (mutex-1)]
                                                                                                   44:
                                                                                                          proc 5 (Th:1) 3.6.3.barrier_nonsol.pml:32 (state 9)
                                                                                                                                                                 [bar[i] = 1]
36:
       proc 2 (Th:1) 3.6.3.barrier nonsol.pml:26 (state 4)
                                                              [temp = count]
                                                                                                                  bar[1] = 0
               Th(2):temp = 3
                                                                                                                  bar[2] = 0
37:
       proc 2 (Th:1) 3.6.3.barrier nonsol.pml:27 (state 5)
                                                              [count = (temp+1)]
                                                                                                                  bar[3] = 0
                                                                                                                  bar[4] = 0
38:
       proc 2 (Th:1) 3.6.3.barrier_nonsol.pml:28 (state 6)
                                                              [mutex = (mutex+1)]
                                                                                                                  bar[5] = 1
                                                                                                          proc 5 (Th:1) 3.6.3.barrier nonsol.pml:33 (state 10) [assert(((((!(bar[1])||:
                                                                                                  (bar[2]))||!(bar[3]))||!(bar[4]))||!(bar[5])))]
39:
       proc 2 (Th:1) 3.6.3.barrier nonsol.pml:29 (state 7)
                                                              \lceil bar[i] = 0 \rceil
                                                                                                          proc 5 (Th:1) 3.6.3.barrier nonsol.pml:34 (state 11) [barrier = (barrier+1)]
                                                                                                                  barrier = 1
40:
       proc 1 (Th:1) 3.6.3.barrier nonsol.pml:25 (state 1)
                                                              [((mutex>0))]
                                                                                                          proc 5 (Th:1) 3.6.3.barrier nonsol.pml:37 (state 15)
                                                                                                                                                                [((barrier>0))]
       proc 1 (Th:1) 3.6.3.barrier nonsol.pml:25 (state 2)
                                                              [mutex = (mutex-1)]
                                                                                                   47:
                                                                                                          proc 5 (Th:1) 3.6.3.barrier nonsol.pml:37 (state 16)
40:
                                                                                                                                                                [barrier = (barrier-1)]
               mutex = 0
                                                                                                                  barrier = 0
       proc 1 (Th:1) 3.6.3.barrier nonsol.pml:26 (state 4)
41:
                                                              [temp = count]
                                                                                                                            Th(5): count = 5
               Th(1):temp = 4
                                                                                                   48:
                                                                                                          proc 5 (Th:1) 3.6.3.barrier nonsol.pml:38 (state 18) [printf('Th(%d): count = %d\\
       proc 1 (Th:1) 3.6.3.barrier_nonsol.pml:27 (state 5)
42:
                                                              [count = (temp+1)]
                                                                                                  n',i,count)]
                                                                                                   49: proc 5 terminates
               count = 5
                                                                                         49
                                                                                                                                                                                            50
  INF646 Métodos Formales
                                                                                                     INF646 Métodos Formales
                                  VK, 2018 - The Little Book of Semaphores
                                                                                                                                     VK, 2018 - The Little Book of Semaphores
  3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                                     3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
       proc 4 (Th:1) 3.6.3.barrier nonsol.pml:31 (state 8)
                                                              [((count==5))]
                                                                                                   57:
                                                                                                          proc 3 (Th:1) 3.6.3.barrier nonsol.pml:31 (state 8)
                                                                                                                                                                 [((count==5))]
       proc 4 (Th:1) 3.6.3.barrier_nonsol.pml:32 (state 9)
                                                              [bar[i] = 1]
                                                                                                          proc 3 (Th:1) 3.6.3.barrier_nonsol.pml:32 (state 9)
                                                                                                                                                                 [bar[i] = 1]
               bar[0] = 0
                                                                                                                  bar[0] = 0
               bar[1] = 0
                                                                                                                  bar[1] = 0
               bar[2] = 0
                                                                                                                  bar[2] = 0
               bar[3] = 0
                                                                                                                  bar[3] = 1
               bar[4] = 1
                                                                                                                  bar[4] = 1
               bar[5] = 1
                                                                                                                  bar[5] = 1
                                                                                                          proc 3 (Th:1) 3.6.3.barrier_nonsol.pml:33 (state 10) [assert(((((!(bar[1])||!
       proc 4 (Th:1) 3.6.3.barrier nonsol.pml:33 (state 10) [assert(((((!(bar[1])||:
(bar[2]))||!(bar[3]))||!(bar[4]))||!(bar[5])))]
                                                                                                  (bar[2]))||!(bar[3]))||!(bar[4]))||!(bar[5])))]
       proc 4 (Th:1) 3.6.3.barrier_nonsol.pml:34 (state 11) [barrier = (barrier+1)]
                                                                                                          proc 3 (Th:1) 3.6.3.barrier_nonsol.pml:34 (state 11) [barrier = (barrier+1)]
               barrier = 1
                                                                                                                  barrier = 1
       proc 4 (Th:1) 3.6.3.barrier_nonsol.pml:37 (state 15) [((barrier>0))]
                                                                                                          proc 3 (Th:1) 3.6.3.barrier nonsol.pml:37 (state 15) [((barrier>0))]
       proc 4 (Th:1) 3.6.3.barrier nonsol.pml:37 (state 16) [barrier = (barrier-1)]
                                                                                                          proc 3 (Th:1) 3.6.3.barrier nonsol.pml:37 (state 16) [barrier = (barrier-1)]
               barrier = 0
                                                                                                                  barrier = 0
                     Th(4): count = 5
                                                                                                                    Th(3): count = 5
       proc 4 (Th:1) 3.6.3.barrier nonsol.pml:38 (state 18) [printf('Th(%d): count = %d\\
                                                                                                   62:
                                                                                                          proc 3 (Th:1) 3.6.3.barrier nonsol.pml:38 (state 18) [printf('Th(%d): count = %d\\
n',i,count)]
                                                                                                  n',i,count)]
56: proc 4 terminates
                                                                                                   63: proc 3 terminates
                                                                                                  . . .
  INF646 Métodos Formales
                                                                                                     INF646 Métodos Formales
                                                                                                                                                                                            52
                                  VK, 2018 - The Little Book of Semaphores
                                                                                         51
                                                                                                                                     VK, 2018 - The Little Book of Semaphores
```

3.6.2 Barrier non-solution (3.6.3.barrier\_nonsol.pml)

```
3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                                 3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
64:
       proc 2 (Th:1) 3.6.3.barrier nonsol.pml:31 (state 8)
                                                            [((count==5))]
                                                                                                      proc 1 (Th:1) 3.6.3.barrier nonsol.pml:28 (state 6)
                                                                                                                                                          [mutex = (mutex+1)]
       proc 2 (Th:1) 3.6.3.barrier_nonsol.pml:32 (state 9)
                                                           [bar[i] = 1]
                                                                                              72:
                                                                                                      proc 1 (Th:1) 3.6.3.barrier nonsol.pml:29 (state 7)
                                                                                                                                                          [bar[i] = 0]
              bar[1] = 0
                                                                                                     proc 1 (Th:1) 3.6.3.barrier nonsol.pml:31 (state 8)
                                                                                                                                                          [((count==5))]
              bar[2] = 1
                                                                                               74:
                                                                                                     proc 1 (Th:1) 3.6.3.barrier_nonsol.pml:32 (state 9)
                                                                                                                                                          [bar[i] = 1]
              bar[3] = 1
                                                                                                             bar[0] = 0
              bar[4] = 1
                                                                                                             bar[1] = 1
               bar[5] = 1
                                                                                                             bar[2] = 1
       proc 2 (Th:1) 3.6.3.barrier nonsol.pml:33 (state 10) [assert(((((!(bar[1])||!
                                                                                                             bar[3] = 1
(bar[2]))||!(bar[3]))||!(bar[4]))||!(bar[5])))]
                                                                                                             bar[4] = 1
      proc 2 (Th:1) 3.6.3.barrier nonsol.pml:34 (state 11) [barrier = (barrier+1)]
                                                                                                             bar[5] = 1
              barrier = 1
                                                                                              spin: 3.6.3.barrier nonsol.pml:33, Error: assertion violated
       proc 2 (Th:1) 3.6.3.barrier_nonsol.pml:37 (state 15) [((barrier>0))]
                                                                                              spin: text of failed assertion: assert(((((!(bar[1])||!(bar[2]))||!(bar[3]))||!(bar[4]))||!
       proc 2 (Th:1) 3.6.3.barrier nonsol.pml:37 (state 16) [barrier = (barrier-1)]
                                                                                              (bar[5])))
              barrier = 0
                                                                                                     proc 1 (Th:1) 3.6.3.barrier_nonsol.pml:33 (state 10) [assert(((((!(bar[1])||:
             Th(2): count = 5
                                                                                              (bar[2]))||!(bar[3]))||!(bar[4]))||!(bar[5])))]
      proc 2 (Th:1) 3.6.3.barrier nonsol.pml:38 (state 18) [printf('Th(%d): count = %d\\
                                                                                              spin: trail ends after 75 steps
n',i,count)]
70: proc 2 terminates
                                                                                      53
                                                                                                                                                                                    54
  INF646 Métodos Formales
                                VK, 2018 - The Little Book of Semaphores
                                                                                                 INF646 Métodos Formales
                                                                                                                               VK, 2018 - The Little Book of Semaphores
  3.6.2 Barrier non-solution (3.6.3.barrier_nonsol.pml)
                                                                                                 3.6.4 Barrier solution (3.6.4a.barrier_sol.pml)
                                                                                              $ cat -n 3.6.4a.barrier_sol.pml | expand
#processes: 2
               count = 5
                                                                                                       /* The Little Book of Semaphores (2.2.1)
               mutex = 1
               barrier = 0
                                                                                                            by A. Downey
               bar[0] = 0
                                                                                                    3
               bar[1] = 1
                                                                                                            Chapter 3. Basic synchronization patterns
               bar[2] = 1
              bar[3] = 1
              bar[4] = 1
                                                                                                            3.6 Barrier
              bar[5] = 1
                                                                                                            3.6.4 Barrier solution
       proc 1 (Th:1) 3.6.3.barrier_nonsol.pml:34 (state 11)
       proc 0 (:init::1) 3.6.3.barrier_nonsol.pml:51 (state 11) <valid end state>
                                                                                                            vk, 2017
6 processes created
                                                                                                   10
                                                                                                   11
                                                                                                                                 /* value for threads number */
                                                                                                       #define THREADS 5
                                                                                                       #define N
                                                                                                                                /* value for barrier limit */
                                                                                                   14
                                                                                                   15 #define wait(sem)
                                                                                                                               atomic { sem > 0; sem-- }
                                                                                                   16
                                                                                                       #define signal(sem) sem++
                                                                                                  17
  INF646 Métodos Formales
                                                                                      55
                                                                                                 INF646 Métodos Formales
                                                                                                                                                                                    56
                                VK, 2018 - The Little Book of Semaphores
                                                                                                                               VK, 2018 - The Little Book of Semaphores
```

```
3.6.4 Barrier solution (3.6.4a.barrier_sol.pml)
                                                                                   3.6.4 Barrier solution (3.6.4a.barrier_sol.pml)
                                          /* barrier is locked */
   18 byte count=0, mutex=1, barrier=0
   19
                                                                                     40
                                                                                        init {
    20
       proctype Th(byte i) {
                                                                                     41
                                                                                             byte i
            byte temp
    21
                                                                                     42
   22
                                                                                    43
                                                                                             atomic {
                                                                                                 for (i: 1 .. THREADS) {
   23
            do
                                                                                    44
            :: wait(mutex)
   24
                                                                                    45
                                                                                                     run Th(i)
   25
                                                                                    46
                    temp=count
                    count=temp+1
                                                                                    47
    26
   27
                signal(mutex)
                                                                                    48
                                                                                             nr pr == 1 ->
   28
                                                                                    49
                                                                                                 printf("barrier = %d\n",barrier)
    29
                :: count == N ->
                                                                                     50
                                                                                    51 }
    30
                    signal(barrier)
    31
                :: else
    32
                fi
    33
                wait(barrier)
                printf("Th(%d): count = %d\n",i,count)
    34
    35
                signal(barrier)
                break /* one only iteration */
    36
    37
            od
    38 }
    39
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                         57
                                                                                   INF646 Métodos Formales
                                                                                                                                                          58
                                                                                                             VK, 2018 - The Little Book of Semaphores
  3.6.4 Barrier solution (3.6.4a.barrier_sol.pml)
                                                                                   3.6.4 Barrier solution (3.6.4a.barrier_sol.pml)
$ spin 3.6.4a.barrier_sol.pml | expand
                                                                                $ spin 3.6.4a.barrier_sol.pml | expand
                  Th(3): count = 5
                                                                                                       Th(4): count = 5
                      Th(4): count = 5
                                                                                                   Th(3): count = 5
              Th(2): count = 5
                                                                                               Th(2): count = 5
                          Th(5): count = 5
                                                                                                           Th(5): count = 5
          Th(1): count = 5
                                                                                           Th(1): count = 5
     barrier = 1
                                                                                      barrier = 2
6 processes created
                                                                                6 processes created
$ spin 3.6.4a.barrier_sol.pml | expand
                      Th(4): count = 5
                  Th(3): count = 5
          Th(1): count = 5
              Th(2): count = 5
                          Th(5): count = 5
     barrier = 1
6 processes created
  INF646 Métodos Formales
                            VK, 2018 - The Little Book of Semaphores
                                                                         59
                                                                                   INF646 Métodos Formales
                                                                                                             VK, 2018 - The Little Book of Semaphores
                                                                                                                                                          60
```

```
3.6.4 Barrier solution (3.6.4b.barrier_sol.pml)
                                                                                                 3.6.4 Barrier solution (3.6.4b.barrier_sol.pml)
                                                                                              $ spin -run 3.6.4b.barrier sol.pml | expand
                                                                                              pan:1: assertion violated ((0<barrier)&&(barrier<5)) (at depth 70)</pre>
    40
       init {
    41
              byte i
                                                                                              pan: wrote 3.6.4b.barrier_sol.pml.trail
    42
    43
                                                                                              (Spin Version 6.4.8 -- 2 March 2018)
             atomic {
    44
                   for (i: 1 .. THREADS) {
                                                                                              Warning: Search not completed
                                                                                                        + Partial Order Reduction
    45
                        run Th(i)
    46
    47
             }
                                                                                              Full statespace search for:
    48
                                                                                                        never claim
              nr pr == 1 ->

    (none specified)

    49
                                                                                                        assertion violations
                  assert(0 < barrier && barrier < 5)</pre>
    50
                  printf("barrier = %d\n",barrier)
                                                                                                        cycle checks

    (disabled by -DSAFETY)

    51 }
                                                                                                        invalid end states
                                                                                              State-vector 64 byte, depth reached 72, errors: 1
                                                                                      61
                                                                                                                                                                                     62
  INF646 Métodos Formales
                                                                                                 INF646 Métodos Formales
                                 VK, 2018 - The Little Book of Semaphores
                                                                                                                                VK, 2018 - The Little Book of Semaphores
  3.6.4 Barrier solution (3.6.4b.barrier_sol.pml)
                                                                                                 3.6.4 Barrier solution (3.6.4b.barrier_sol.pml)
                                                                                               24:
                                                                                                      proc 4 (Th:1) 3.6.4b.barrier_sol.pml:24 (state 1)
                                                                                                                                                            [((mutex>0))]
$ spin -t -p -q -l 3.6.4b.barrier sol.pml | expand
                                                                                               24:
                                                                                                      proc 4 (Th:1) 3.6.4b.barrier sol.pml:24 (state 2)
                                                                                                                                                            [mutex = (mutex-1)]
using statement merging
                                                                                                             mutex = 0
      proc 0 (:init::1) 3.6.4b.barrier sol.pml:44 (state 1) [i = 1]
                                                                                               25:
                                                                                                      proc 4 (Th:1) 3.6.4b.barrier sol.pml:25 (state 4)
                                                                                                                                                            [temp = count]
               :init:(0):i = 1
                                                                                                             Th(4):temp = 1
      proc 0 (:init::1) 3.6.4b.barrier_sol.pml:44 (state 2) [((i<=5))]</pre>
                                                                                               26:
                                                                                                      proc 4 (Th:1) 3.6.4b.barrier_sol.pml:26 (state 5)
                                                                                                                                                            [count = (temp+1)]
Starting Th with pid 1
                                                                                                             count = 2
      proc 0 (:init::1) 3.6.4b.barrier_sol.pml:45 (state 3) [(run Th(i))]
 3:
                                                                                               27:
                                                                                                      proc 4 (Th:1) 3.6.4b.barrier_sol.pml:27 (state 6)
                                                                                                                                                            [mutex = (mutex+1)]
       proc 0 (:init::1) 3.6.4b.barrier sol.pml:44 (state 4) [i = (i+1)]
                                                                                                             mutex = 1
               :init:(0):i = 2
      proc 0 (:init::1) 3.6.4b.barrier_sol.pml:44 (state 2) [((i<=5))]</pre>
                                                                                               28:
                                                                                                      proc 3 (Th:1) 3.6.4b.barrier_sol.pml:24 (state 1)
                                                                                                                                                            [((mutex>0))]
Starting Th with pid 2
                                                                                               28:
                                                                                                      proc 3 (Th:1) 3.6.4b.barrier_sol.pml:24 (state 2)
                                                                                                                                                            [mutex = (mutex-1)]
                                                                                                             mutex = 0
               :init:(0):i = 6
                                                                                                      proc 3 (Th:1) 3.6.4b.barrier_sol.pml:25 (state 4)
                                                                                               29:
                                                                                                                                                            [temp = count]
17:
       proc 0 (:init::1) 3.6.4b.barrier sol.pml:46 (state 5)
                                                            [else]
                                                                                                              Th(3):temp = 2
18:
       proc 0 (:init::1) 3.6.4b.barrier_sol.pml:46 (state 6)
                                                             [qoto:b1]
                                                                                               30:
                                                                                                      proc 3 (Th:1) 3.6.4b.barrier_sol.pml:26 (state 5)
                                                                                                                                                            [count = (temp+1)]
19:
       proc 0 (:init::1) 3.6.4b.barrier sol.pml:46 (state 9)
                                                             [break]
                                                                                                              count = 3
                                                                                               31:
                                                                                                      proc 3 (Th:1) 3.6.4b.barrier sol.pml:27 (state 6)
                                                                                                                                                            [mutex = (mutex+1)]
       proc 5 (Th:1) 3.6.4b.barrier_sol.pml:24 (state 1)
20:
                                                             [((mutex>0))]
                                                                                                             mutex = 1
       proc 5 (Th:1) 3.6.4b.barrier sol.pml:24 (state 2)
                                                             [mutex = (mutex-1)]
               mutex = 0
                                                                                                      proc 2 (Th:1) 3.6.4b.barrier sol.pml:24 (state 1)
                                                                                                                                                            [((mutex>0))]
                                                                                               32:
       proc 5 (Th:1) 3.6.4b.barrier_sol.pml:25 (state 4)
21:
                                                             [temp = count]
                                                                                               32:
                                                                                                      proc 2 (Th:1) 3.6.4b.barrier sol.pml:24 (state 2)
                                                                                                                                                            [mutex = (mutex-1)]
               Th(5):temp = 0
                                                                                                             mutex = 0
22:
       proc 5 (Th:1) 3.6.4b.barrier_sol.pml:26 (state 5)
                                                             [count = (temp+1)]
                                                                                               33:
                                                                                                      proc 2 (Th:1) 3.6.4b.barrier_sol.pml:25 (state 4)
                                                                                                                                                            [temp = count]
               count = 1
                                                                                                              Th(2):temp = 3
23:
       proc 5 (Th:1) 3.6.4b.barrier sol.pml:27 (state 6)
                                                             [mutex = (mutex+1)]
                                                                                               34:
                                                                                                      proc 2 (Th:1) 3.6.4b.barrier sol.pml:26 (state 5)
                                                                                                                                                            [count = (temp+1)]
               mutex = 1
                                                                                                              count = 4
                                                                                                      proc 2 (Th:1) 3.6.4b.barrier_sol.pml:27 (state 6)
                                                                                                                                                            [mutex = (mutex+1)]
                                                                                                             mutex = 1
```

63

**INF646 Métodos Formales** 

VK, 2018 - The Little Book of Semaphores

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

#### 3.6.4 Barrier solution (3.6.4b.barrier\_sol.pml) 3.6.4 Barrier solution (3.6.4b.barrier\_sol.pml) proc 1 (Th:1) 3.6.4b.barrier\_sol.pml:24 (state 1) [((mutex>0))] 45: proc 4 (Th:1) 3.6.4b.barrier sol.pml:29 (state 7) 36: [((count==5))] [mutex = (mutex-1)] [barrier = (barrier+1)] 36: proc 1 (Th:1) 3.6.4b.barrier\_sol.pml:24 (state 2) 46: proc 4 (Th:1) 3.6.4b.barrier\_sol.pml:30 (state 8) mutex = 0barrier = 237: proc 1 (Th:1) 3.6.4b.barrier sol.pml:25 (state 4) [temp = count] 47: proc 4 (Th:1) 3.6.4b.barrier sol.pml:33 (state 12) [((barrier>0))] 47: proc 4 (Th:1) 3.6.4b.barrier\_sol.pml:33 (state 13) [barrier = (barrier-1)] Th(1):temp = 438: proc 1 (Th:1) 3.6.4b.barrier sol.pml:26 (state 5) [count = (temp+1)]barrier = 1count = 5Th(4): count = 5 proc 4 (Th:1) 3.6.4b.barrier\_sol.pml:34 (state 15) [printf('Th(%d): count = %d\\ proc **5** (Th:1) 3.6.4b.barrier sol.pml:29 (state 7) [((count==5))] 39: n',i,count)] 40: proc 5 (Th:1) 3.6.4b.barrier\_sol.pml:30 (state 8) [barrier = (barrier+1)] 49: proc 4 (Th:1) 3.6.4b.barrier\_sol.pml:35 (state 16) [barrier = (barrier+1)] barrier = 1barrier = 241: proc **5** (Th:1) 3.6.4b.barrier sol.pml:33 (state 12) [((barrier>0))] 50: proc 4 terminates [barrier = (barrier-1)] 41: proc **5** (Th:1) 3.6.4b.barrier sol.pml:33 (state 13) barrier = 0proc 3 (Th:1) 3.6.4b.barrier\_sol.pml:29 (state 7) [((count==5))] Th(5): count = 5 proc **5** (Th:1) 3.6.4b.barrier sol.pml:34 (state 15) [printf('Th(%d): count = %d\\ 52: proc **3** (Th:1) 3.6.4b.barrier sol.pml:30 (state 8) [barrier = (barrier+1)] n',i,count)] barrier = 3proc 5 (Th:1) 3.6.4b.barrier sol.pml:35 (state 16) [barrier = (barrier+1)] 53: proc 3 (Th:1) 3.6.4b.barrier sol.pml:33 (state 12) [((barrier>0))] barrier = 153: proc **3** (Th:1) 3.6.4b.barrier sol.pml:33 (state 13) [barrier = (barrier-1)] 44: proc 5 terminates barrier = 2Th(3): count = 5 54: proc **3** (Th:1) 3.6.4b.barrier sol.pml:34 (state 15) [printf('Th(%d): count = %d\\ n',i,count)] proc 3 (Th:1) 3.6.4b.barrier\_sol.pml:35 (state 16) [barrier = (barrier+1)] barrier = 356: proc 3 terminates 66 **INF646 Métodos Formales** 65 VK, 2018 - The Little Book of Semaphores **INF646 Métodos Formales** VK, 2018 - The Little Book of Semaphores 3.6.4 Barrier solution (3.6.4b.barrier\_sol.pml) 3.6.4 Barrier solution (3.6.4b.barrier\_sol.pml)

```
[((count==5))]
57:
       proc 2 (Th:1) 3.6.4b.barrier sol.pml:29 (state 7)
       proc 2 (Th:1) 3.6.4b.barrier_sol.pml:30 (state 8)
                                                                [barrier = (barrier+1)]
58:
               barrier = 4
       proc 2 (Th:1) 3.6.4b.barrier sol.pml:33 (state 12)
                                                                [((barrier>0))]
59:
       proc 2 (Th:1) 3.6.4b.barrier sol.pml:33 (state 13)
                                                                [barrier = (barrier-1)]
59:
               barrier = 3
              Th(2): count = 5
       proc 2 (Th:1) 3.6.4b.barrier sol.pml:34 (state 15)
                                                               [printf('Th(%d): count = %d\\
n',i,count)]
       proc 2 (Th:1) 3.6.4b.barrier sol.pml:35 (state 16)
                                                               [barrier = (barrier+1)]
               barrier = 4
62: proc 2 terminates
63:
       proc 1 (Th:1) 3.6.4b.barrier sol.pml:27 (state 6)
                                                               [mutex = (mutex+1)]
               mutex = 1
64:
       proc 1 (Th:1) 3.6.4b.barrier_sol.pml:29 (state 7)
                                                               [((count==5))]
65:
       proc 1 (Th:1) 3.6.4b.barrier_sol.pml:30 (state 8)
                                                                [barrier = (barrier+1)]
               barrier = 5
       proc 1 (Th:1) 3.6.4b.barrier sol.pml:33 (state 12)
                                                               [((barrier>0))]
66:
       proc 1 (Th:1) 3.6.4b.barrier_sol.pml:33 (state 13)
                                                               [barrier = (barrier-1)]
66:
               barrier = 4
         Th(1): count = 5
67:
       proc 1 (Th:1) 3.6.4b.barrier_sol.pml:34 (state 15)
                                                               [printf('Th(%d): count = %d\\
n',i,count)]
       proc 1 (Th:1) 3.6.4b.barrier_sol.pml:35 (state 16)
                                                               [barrier = (barrier+1)]
               barrier = 5
69: proc 1 terminates
```

```
proc 0 (:init::1) 3.6.4b.barrier sol.pml:48 (state 11) [(( nr pr==1))]
70:
spin: 3.6.4b.barrier_sol.pml:49, Error: assertion violated
spin: text of failed assertion: assert(((0<barrier)&&(barrier<5)))</pre>
        proc 0 (:init::1) 3.6.4b.barrier_sol.pml:49 (state 12)
[assert(((0<barrier)&&(barrier<5)))]
spin: trail ends after 71 steps
#processes: 1
                count = 5
                mutex = 1
                barrier = 5
       proc 0 (:init::1) 3.6.4b.barrier sol.pml:50 (state 13)
6 processes created
```

#### 3.6.5 Bad barrier solution (3.6.5.bad\_barrier.pml) 3.6.5 Bad barrier solution (3.6.5.bad\_barrier.pml) proctype Th(byte i) { \$ cat -n 3.6.5.bad\_barrier.pml | expand byte temp 22 /\* The Little Book of Semaphores (2.2.1) 23 rendezvous: by A. Downey 24 do 3 25 :: wait(mutex) Chapter 3. Basic synchronization patterns 4 26 temp=count 5 27 count=temp+1 6 3.6 Barrier 28 if 3.6.5 Bad barrier solution (deadlock) 29 :: count == N -> 8 30 signal(barrier) vk, 2017 9 31 :: else \*/ 10 32 11 33 wait(barrier) /\* value for threads number \*/ #define THREADS 3 12 34 printf("Th(%d): count = %d\n",i,count) /\* value for barrier limit \*/ #define N 13 35 signal(barrier) 14 36 signal(mutex) #define wait(sem) atomic { sem > 0; sem-- } 37 break /\* one only iteration \*/ #define signal(sem) sem++ 16 38 od 17 39 critical\_point: byte count=0, mutex=1, barrier=0 /\* barrier is locked \*/ 40 19 . . .

69

INF646 Métodos Formales

VK, 2018 - The Little Book of Semaphores

# 3.6.5 Bad barrier solution (3.6.5.bad\_barrier.pml)

**INF646 Métodos Formales** 

```
41
    42 init {
            byte i
    43
    44
   45
            atomic {
    46
                for (i: 1 .. THREADS) {
   47
                    run Th(i)
    48
    49
    50 }
$ spin 3.6.5.bad_barrier.pml | expand
     timeout
#processes: 4
                count = 1
                mutex = 0
                barrier = 0
19:
        proc 3 (Th:1) 3.6.5.bad_barrier.pml:24 (state 18)
       proc 2 (Th:1) 3.6.5.bad barrier.pml:24 (state 18)
19:
       proc 1 (Th:1) 3.6.5.bad_barrier.pml:33 (state 13)
19:
       proc 0 (:init::1) 3.6.5.bad_barrier.pml:50 (state 11) <valid end
19:
state>
4 processes created
  INF646 Métodos Formales
```

VK, 2018 - The Little Book of Semaphores

VK, 2018 - The Little Book of Semaphores

71