Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Верификация анализ программ

Отчет по лабораторной работе №2
Проверка модели семафора

Работу выполнил:
Тарасов С.И.
Группа:
3540901/91502
Преподаватель:
Ицыксон В.М.

Санкт-Петербург
2020

Содержание

1. Цель работы

Составить при помощи NuSMV модель программы с 5-ю пользователями и ресурсом, который защищён семафором. Проверить свойства при помощи LTL формул:

1) При инициализации семафора значением 1 убедиться, что в критической секции не может быть больше одного процесса

2) Убедиться, что каждый процесс в конце концов получит доступ к критической секции

3) При инициализации семафора значением 3 убедиться, что в критической секции может быть 1, 2 или 3 процесса

4) При инициализации семафора значением 3 убедиться, что в критической секции не может быть 4 процесса

2. Теоретическая информация

Семафор - инструмент синхронизации доступа к ресурсу при помощи специальных токенов, максимальное количество которых задаётся при инициализации. От количества токенов зависит максимальное число потоков, которые могут использовать ресурс одновременно. При отсутствии токенов, семафор блокирует все процессы до высвобождения токенов.

3. Ход выполнения работы

Для моделирования программы использовался NuSMV - расширение SMV, которое производит проверку моделей на соответствие LTL или CTL формулам, которые выражают некоторые свойства программы. Программа представляется в виде конечного автомата.

Программа моделирования представлена на листинге.

Результат генерации.

Листинг 1: Программа моделирования

```
MODULE main
 VAR
    semaphore    : {0,1,2,3,4,5,6,7,8,9,10};
    apr         :0 .. 4                    ;
    proc0       : process user(semaphore,0,apr);
    proc1       : process user(semaphore,1,apr);
    proc2       : process user(semaphore,2,apr);
    proc3       : process user(semaphore,3,apr);
    proc4       : process user(semaphore,4,apr);
 ASSIGN
    init(semaphore) := 1;
    init(apr) :=0;


LTLSPEC (F (proc0.state = critical2) & F (proc1.state = critical2) & F (proc2.
    ↪ state = critical2) & F (proc3.state = critical2) & F (proc4.state =
    ↪ critical2))
LTLSPEC (! F (proc0.state = critical2 & proc1.state = idle & proc2.state = idle
    ↪ & proc3.state = idle & proc4.state = idle))
LTLSPEC (! F (proc0.state = critical2 & proc1.state = critical2 & proc2.state =
    ↪ idle & proc3.state = idle & proc4.state = idle))
LTLSPEC (! F (proc0.state = critical2 & proc1.state = critical2 & proc2.state =
    ↪ critical2 & proc3.state = idle & proc4.state = idle))
```

```
19 LTLSPEC (! F (proc0.state = critical2 & proc1.state = critical2 & proc2.state =
     ↪ critical2 & proc3.state = critical2 & proc4.state = idle))
20 LTLSPEC (! F (proc0.state = critical2 & proc1.state = critical2 & proc2.state =
     ↪ critical2 & proc3.state = critical2 & proc4.state = critical2))
21
22
23 MODULE user(semaphore, pNum, apr)
24  VAR
25    state : {idle, enqueue, critical1, critical2, exiting};
26  ASSIGN
27    init(state) := idle;
28    next(state) :=
29      case
30        state = idle                                                    :
     ↪ enqueue;
31        state = enqueue & semaphore >0 & apr = pNum                    :
     ↪ critical1;
32        state = critical1                                               :
     ↪ critical2;
33        state = critical2                                               :
     ↪ exiting;
34        state = exiting                                                 :
     ↪ idle;
35        TRUE                                                            :
     ↪ state;
36      esac;
37
38    next(semaphore) :=
39      case
40        state = enqueue & semaphore >0 & apr = pNum  : semaphore -1;
41        state = exiting & semaphore < 1              : semaphore +1;
42        TRUE                                         : semaphore;
43      esac;
44
45    next(apr) :=
46        case
47          state = enqueue & semaphore >0 & apr = pNum & apr < 4      : apr +1;
48          state = enqueue & semaphore >0 & apr = pNum & apr = 4      :0;
49          TRUE                                                        : apr;
50        esac;
51  FAIRNESS
52    running
```

На строках 15 - 20 выписаны LTL формулы. Первая формула проверяет, что в все потоки в конечном счёте побывают в критической секции. Вторая формула, что невозможно наличие 1 потока в критической секции. Третья - 2 потока. Четвертая - 3 потока. Пятая - 4 потока. Шестая - 5 потоков. В рамках семафора так же организована очередь с приоритетом (переменная apr) для того, чтобы ни один из потоков был заблокирован вечно.

При инициализации семафора значениями от 0 до 5 выходят результаты, представленные в примечании. При отрицательном результате проверки формулы выписывается контрпример.

4. Выводы

В рамках работы была создана программа по построению АСТ для Java кода. На данный момент программа поддерживает только базовые конструкции. Полное корректное

парсирование возможно с использованием специальных инструментов.

5. Приложения

5.1. Приложение А

Результаты моделирования при инициализации семафора нулём.

Листинг 2: Инициализация семаформа - 0

```
*** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:56 2015)
*** Enabled addons are: compass
*** For more information on NuSMV see <http://nusmv.fbk.eu>
*** or email to <nusmv-users@list.fbk.eu>.
*** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>

*** Copyright (c) 2010-2014, Fondazione Bruno Kessler

*** This version of NuSMV is linked to the CUDD library version 2.4.1
*** Copyright (c) 1995-2004, Regents of the University of Colorado

*** This version of NuSMV is linked to the MiniSat SAT solver.
*** See http://minisat.se/MiniSat.html
*** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
*** Copyright (c) 2007-2010, Niklas Sorensson

WARNING *** The model contains PROCESSes or ISAs. ***
WARNING *** The HRC hierarchy will not be usable. ***
-- specification ((((( F proc0.state = critical2 & F proc1.state = critical2) &
   ↪ F proc2.state = critical2) & F proc3.state = critical2) & F proc4.state
   ↪ = critical2) is false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
Trace Type: Counterexample
  -> State: 1.1 <-
    semaphore = 0
    apr = 0
    proc0.state = idle
    proc1.state = idle
    proc2.state = idle
    proc3.state = idle
    proc4.state = idle
  -> Input: 1.2 <-
    _process_selector_ = proc0
    running = FALSE
    proc4.running = FALSE
    proc3.running = FALSE
    proc2.running = FALSE
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 1.2 <-
    proc0.state = enqueue
  -> Input: 1.3 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 1.3 <-
    proc1.state = enqueue
  -> Input: 1.4 <-
    _process_selector_ = proc2
```

```
49      proc2.running = TRUE
50      proc1.running = FALSE
51    -> State: 1.4 <-
52      proc2.state = enqueue
53    -> Input: 1.5 <-
54      _process_selector_ = proc3
55      proc3.running = TRUE
56      proc2.running = FALSE
57    -> State: 1.5 <-
58      proc3.state = enqueue
59    -> Input: 1.6 <-
60      _process_selector_ = proc4
61      proc4.running = TRUE
62      proc3.running = FALSE
63    -- Loop starts here
64    -> State: 1.6 <-
65      proc4.state = enqueue
66    -> Input: 1.7 <-
67      _process_selector_ = main
68      running = TRUE
69      proc4.running = FALSE
70    -- Loop starts here
71    -> State: 1.7 <-
72    -> Input: 1.8 <-
73      _process_selector_ = proc0
74      running = FALSE
75      proc0.running = TRUE
76    -- Loop starts here
77    -> State: 1.8 <-
78    -> Input: 1.9 <-
79      _process_selector_ = proc1
80      proc1.running = TRUE
81      proc0.running = FALSE
82    -- Loop starts here
83    -> State: 1.9 <-
84    -> Input: 1.10 <-
85      _process_selector_ = proc2
86      proc2.running = TRUE
87      proc1.running = FALSE
88    -- Loop starts here
89    -> State: 1.10 <-
90    -> Input: 1.11 <-
91      _process_selector_ = proc3
92      proc3.running = TRUE
93      proc2.running = FALSE
94    -- Loop starts here
95    -> State: 1.11 <-
96    -> Input: 1.12 <-
97      _process_selector_ = proc4
98      proc4.running = TRUE
99      proc3.running = FALSE
100   -- Loop starts here
101   -> State: 1.12 <-
102   -> Input: 1.13 <-
103     _process_selector_ = main
104     running = TRUE
105     proc4.running = FALSE
106   -> State: 1.13 <-
107 -- specification !( F ((((proc0.state = critical2 & proc1.state = idle) & proc2.
        ↪ state = idle) & proc3.state = idle) & proc4.state = idle))  is true
```

```
108 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    ↪ proc2.state = idle) & proc3.state = idle) & proc4.state = idle))  is true
109 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    ↪ proc2.state = critical2) & proc3.state = idle) & proc4.state = idle))  is
    ↪ true
110 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state = idle))
    ↪   is true
111 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state =
    ↪ critical2))  is true
```

5.2. Приложение B

Результаты моделирования при инициализации семафора 1-й.

Листинг 3: Инициализация семаформа - 1

```
1  *** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:56 2015)
2  *** Enabled addons are: compass
3  *** For more information on NuSMV see <http://nusmv.fbk.eu>
4  *** or email to <nusmv-users@list.fbk.eu>.
5  *** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>
6
7  *** Copyright (c) 2010-2014, Fondazione Bruno Kessler
8
9  *** This version of NuSMV is linked to the CUDD library version 2.4.1
10 *** Copyright (c) 1995-2004, Regents of the University of Colorado
11
12 *** This version of NuSMV is linked to the MiniSat SAT solver.
13 *** See http://minisat.se/MiniSat.html
14 *** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
15 *** Copyright (c) 2007-2010, Niklas Sorensson
16
17 WARNING *** The model contains PROCESSes or ISAs. ***
18 WARNING *** The HRC hierarchy will not be usable. ***
19 -- specification ((( F proc0.state = critical2 &  F proc1.state = critical2) &
    ↪  F proc2.state = critical2) &  F proc3.state = critical2) &  F proc4.state
    ↪  = critical2)  is true
20 -- specification !( F ((((proc0.state = critical2 & proc1.state = idle) & proc2.
    ↪ state = idle) & proc3.state = idle) & proc4.state = idle))  is false
21 -- as demonstrated by the following execution sequence
22 Trace Description: LTL Counterexample
23 Trace Type: Counterexample
24   -> State: 1.1 <-
25     semaphore = 1
26     apr = 0
27     proc0.state = idle
28     proc1.state = idle
29     proc2.state = idle
30     proc3.state = idle
31     proc4.state = idle
32   -> Input: 1.2 <-
33     _process_selector_ = proc0
34     running = FALSE
35     proc4.running = FALSE
36     proc3.running = FALSE
37     proc2.running = FALSE
38     proc1.running = FALSE
39     proc0.running = TRUE
```

```
-> State: 1.2 <-
  proc0.state = enqueue
-> Input: 1.3 <-
  _process_selector_ = proc1
  proc1.running = TRUE
  proc0.running = FALSE
-> State: 1.3 <-
  proc1.state = enqueue
-> Input: 1.4 <-
  _process_selector_ = proc2
  proc2.running = TRUE
  proc1.running = FALSE
-> State: 1.4 <-
  proc2.state = enqueue
-> Input: 1.5 <-
  _process_selector_ = proc3
  proc3.running = TRUE
  proc2.running = FALSE
-> State: 1.5 <-
  proc3.state = enqueue
-> Input: 1.6 <-
  _process_selector_ = proc4
  proc4.running = TRUE
  proc3.running = FALSE
-> State: 1.6 <-
  proc4.state = enqueue
-> Input: 1.7 <-
  _process_selector_ = proc0
  proc4.running = FALSE
  proc0.running = TRUE
-> State: 1.7 <-
  semaphore = 0
  apr = 1
  proc0.state = critical1
-> Input: 1.8 <-
-> State: 1.8 <-
  proc0.state = critical2
-> Input: 1.9 <-
-> State: 1.9 <-
  proc0.state = exiting
-> Input: 1.10 <-
-> State: 1.10 <-
  semaphore = 1
  proc0.state = idle
-> Input: 1.11 <-
  _process_selector_ = proc1
  proc1.running = TRUE
  proc0.running = FALSE
-> State: 1.11 <-
  semaphore = 0
  apr = 2
  proc1.state = critical1
-> Input: 1.12 <-
-> State: 1.12 <-
  proc1.state = critical2
-> Input: 1.13 <-
-> State: 1.13 <-
  proc1.state = exiting
-> Input: 1.14 <-
-> State: 1.14 <-
```

```
100      semaphore = 1
101      proc1.state = idle
102    -> Input: 1.15 <-
103      _process_selector_ = proc2
104      proc2.running = TRUE
105      proc1.running = FALSE
106    -> State: 1.15 <-
107      semaphore = 0
108      apr = 3
109      proc2.state = critical1
110    -> Input: 1.16 <-
111    -> State: 1.16 <-
112      proc2.state = critical2
113    -> Input: 1.17 <-
114    -> State: 1.17 <-
115      proc2.state = exiting
116    -> Input: 1.18 <-
117    -> State: 1.18 <-
118      semaphore = 1
119      proc2.state = idle
120    -> Input: 1.19 <-
121      _process_selector_ = proc3
122      proc3.running = TRUE
123      proc2.running = FALSE
124    -> State: 1.19 <-
125      semaphore = 0
126      apr = 4
127      proc3.state = critical1
128    -> Input: 1.20 <-
129    -> State: 1.20 <-
130      proc3.state = critical2
131    -> Input: 1.21 <-
132    -> State: 1.21 <-
133      proc3.state = exiting
134    -> Input: 1.22 <-
135    -> State: 1.22 <-
136      semaphore = 1
137      proc3.state = idle
138    -> Input: 1.23 <-
139      _process_selector_ = proc4
140      proc4.running = TRUE
141      proc3.running = FALSE
142    -> State: 1.23 <-
143      semaphore = 0
144      apr = 0
145      proc4.state = critical1
146    -> Input: 1.24 <-
147    -> State: 1.24 <-
148      proc4.state = critical2
149    -> Input: 1.25 <-
150    -> State: 1.25 <-
151      proc4.state = exiting
152    -> Input: 1.26 <-
153    -> State: 1.26 <-
154      semaphore = 1
155      proc4.state = idle
156    -> Input: 1.27 <-
157      _process_selector_ = proc0
158      proc4.running = FALSE
159      proc0.running = TRUE
```

```
  -> State: 1.27 <-
    proc0.state = enqueue
  -> Input: 1.28 <-
  -> State: 1.28 <-
    semaphore = 0
    apr = 1
    proc0.state = critical1
  -> Input: 1.29 <-
  -> State: 1.29 <-
    proc0.state = critical2
  -> Input: 1.30 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc0.running = FALSE
  -> State: 1.30 <-
    proc4.state = enqueue
  -> Input: 1.31 <-
    _process_selector_ = proc1
    proc4.running = FALSE
    proc1.running = TRUE
  -- Loop starts here
  -> State: 1.31 <-
    proc1.state = enqueue
  -> Input: 1.32 <-
    _process_selector_ = proc0
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 1.32 <-
    proc0.state = exiting
  -> Input: 1.33 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 1.33 <-
  -> Input: 1.34 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 1.34 <-
    proc2.state = enqueue
  -> Input: 1.35 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 1.35 <-
    proc3.state = enqueue
  -> Input: 1.36 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
  -> State: 1.36 <-
  -> Input: 1.37 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 1.37 <-
    semaphore = 1
    proc0.state = idle
  -> Input: 1.38 <-
    _process_selector_ = proc1
```

```
220      proc1.running = TRUE
221      proc0.running = FALSE
222    -> State: 1.38 <-
223      semaphore = 0
224      apr = 2
225      proc1.state = critical1
226    -> Input: 1.39 <-
227    -> State: 1.39 <-
228      proc1.state = critical2
229    -> Input: 1.40 <-
230    -> State: 1.40 <-
231      proc1.state = exiting
232    -> Input: 1.41 <-
233    -> State: 1.41 <-
234      semaphore = 1
235      proc1.state = idle
236    -> Input: 1.42 <-
237      _process_selector_ = proc2
238      proc2.running = TRUE
239      proc1.running = FALSE
240    -> State: 1.42 <-
241      semaphore = 0
242      apr = 3
243      proc2.state = critical1
244    -> Input: 1.43 <-
245    -> State: 1.43 <-
246      proc2.state = critical2
247    -> Input: 1.44 <-
248    -> State: 1.44 <-
249      proc2.state = exiting
250    -> Input: 1.45 <-
251    -> State: 1.45 <-
252      semaphore = 1
253      proc2.state = idle
254    -> Input: 1.46 <-
255      _process_selector_ = proc3
256      proc3.running = TRUE
257      proc2.running = FALSE
258    -> State: 1.46 <-
259      semaphore = 0
260      apr = 4
261      proc3.state = critical1
262    -> Input: 1.47 <-
263    -> State: 1.47 <-
264      proc3.state = critical2
265    -> Input: 1.48 <-
266    -> State: 1.48 <-
267      proc3.state = exiting
268    -> Input: 1.49 <-
269    -> State: 1.49 <-
270      semaphore = 1
271      proc3.state = idle
272    -> Input: 1.50 <-
273      _process_selector_ = proc4
274      proc4.running = TRUE
275      proc3.running = FALSE
276    -> State: 1.50 <-
277      semaphore = 0
278      apr = 0
279      proc4.state = critical1
```

```
  -> Input: 1.51 <-
  -> State: 1.51 <-
    proc4.state = critical2
  -> Input: 1.52 <-
  -> State: 1.52 <-
    proc4.state = exiting
  -> Input: 1.53 <-
  -> State: 1.53 <-
    semaphore = 1
    proc4.state = idle
  -> Input: 1.54 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 1.54 <-
    proc0.state = enqueue
  -> Input: 1.55 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 1.55 <-
    proc1.state = enqueue
  -> Input: 1.56 <-
    _process_selector_ = proc0
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 1.56 <-
    semaphore = 0
    apr = 1
    proc0.state = critical1
  -> Input: 1.57 <-
  -> State: 1.57 <-
    proc0.state = critical2
  -> Input: 1.58 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc0.running = FALSE
  -> State: 1.58 <-
    proc4.state = enqueue
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = idle) & proc3.state = idle) & proc4.state = idle))  is true
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = idle) & proc4.state = idle))  is
   ↪ true
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state = idle))
   ↪   is true
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state =
   ↪ critical2))  is true
```

5.3. Приложение C

Результаты моделирования при инициализации семафора 2-й.

Листинг 4: Инициализация семаформа - 2

```
*** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:56 2015)
*** Enabled addons are: compass
*** For more information on NuSMV see <http://nusmv.fbk.eu>
```

```
*** or email to <nusmv-users@list.fbk.eu>.
*** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>

*** Copyright (c) 2010-2014, Fondazione Bruno Kessler

*** This version of NuSMV is linked to the CUDD library version 2.4.1
*** Copyright (c) 1995-2004, Regents of the University of Colorado

*** This version of NuSMV is linked to the MiniSat SAT solver.
*** See http://minisat.se/MiniSat.html
*** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
*** Copyright (c) 2007-2010, Niklas Sorensson

WARNING *** The model contains PROCESSes or ISAs. ***
WARNING *** The HRC hierarchy will not be usable. ***
-- specification ((((  F proc0.state = critical2 &  F proc1.state = critical2) &
   ↪  F proc2.state = critical2) &  F proc3.state = critical2) &  F proc4.state
   ↪  = critical2)  is true
-- specification !( F ((((proc0.state = critical2 & proc1.state = idle) & proc2.
   ↪ state = idle) & proc3.state = idle) & proc4.state = idle))  is false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
Trace Type: Counterexample
  -> State: 1.1 <-
    semaphore = 2
    apr = 0
    proc0.state = idle
    proc1.state = idle
    proc2.state = idle
    proc3.state = idle
    proc4.state = idle
  -> Input: 1.2 <-
    _process_selector_ = proc0
    running = FALSE
    proc4.running = FALSE
    proc3.running = FALSE
    proc2.running = FALSE
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 1.2 <-
    proc0.state = enqueue
  -> Input: 1.3 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 1.3 <-
    proc1.state = enqueue
  -> Input: 1.4 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 1.4 <-
    proc2.state = enqueue
  -> Input: 1.5 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 1.5 <-
    proc3.state = enqueue
  -> Input: 1.6 <-
```

```
61      _process_selector_ = proc4
62      proc4.running = TRUE
63      proc3.running = FALSE
64    -> State: 1.6 <-
65      proc4.state = enqueue
66    -> Input: 1.7 <-
67      _process_selector_ = proc0
68      proc4.running = FALSE
69      proc0.running = TRUE
70    -> State: 1.7 <-
71      semaphore = 1
72      apr = 1
73      proc0.state = critical1
74    -> Input: 1.8 <-
75      _process_selector_ = proc1
76      proc1.running = TRUE
77      proc0.running = FALSE
78    -> State: 1.8 <-
79      semaphore = 0
80      apr = 2
81      proc1.state = critical1
82    -> Input: 1.9 <-
83    -> State: 1.9 <-
84      proc1.state = critical2
85    -> Input: 1.10 <-
86    -> State: 1.10 <-
87      proc1.state = exiting
88    -> Input: 1.11 <-
89    -> State: 1.11 <-
90      semaphore = 1
91      proc1.state = idle
92    -> Input: 1.12 <-
93      _process_selector_ = proc2
94      proc2.running = TRUE
95      proc1.running = FALSE
96    -> State: 1.12 <-
97      semaphore = 0
98      apr = 3
99      proc2.state = critical1
100   -> Input: 1.13 <-
101   -> State: 1.13 <-
102     proc2.state = critical2
103   -> Input: 1.14 <-
104   -> State: 1.14 <-
105     proc2.state = exiting
106   -> Input: 1.15 <-
107   -> State: 1.15 <-
108     semaphore = 1
109     proc2.state = idle
110   -> Input: 1.16 <-
111     _process_selector_ = proc3
112     proc3.running = TRUE
113     proc2.running = FALSE
114   -> State: 1.16 <-
115     semaphore = 0
116     apr = 4
117     proc3.state = critical1
118   -> Input: 1.17 <-
119   -> State: 1.17 <-
120     proc3.state = critical2
```

```
121  -> Input: 1.18 <-
122  -> State: 1.18 <-
123    proc3.state = exiting
124  -> Input: 1.19 <-
125  -> State: 1.19 <-
126    semaphore = 1
127    proc3.state = idle
128  -> Input: 1.20 <-
129    _process_selector_ = proc4
130    proc4.running = TRUE
131    proc3.running = FALSE
132  -> State: 1.20 <-
133    semaphore = 0
134    apr = 0
135    proc4.state = critical1
136  -> Input: 1.21 <-
137  -> State: 1.21 <-
138    proc4.state = critical2
139  -> Input: 1.22 <-
140  -> State: 1.22 <-
141    proc4.state = exiting
142  -> Input: 1.23 <-
143  -> State: 1.23 <-
144    semaphore = 1
145    proc4.state = idle
146  -> Input: 1.24 <-
147    _process_selector_ = proc0
148    proc4.running = FALSE
149    proc0.running = TRUE
150  -> State: 1.24 <-
151    proc0.state = critical2
152  -> Input: 1.25 <-
153    _process_selector_ = proc4
154    proc4.running = TRUE
155    proc0.running = FALSE
156  -> State: 1.25 <-
157    proc4.state = enqueue
158  -> Input: 1.26 <-
159    _process_selector_ = proc1
160    proc4.running = FALSE
161    proc1.running = TRUE
162  -- Loop starts here
163  -> State: 1.26 <-
164    proc1.state = enqueue
165  -> Input: 1.27 <-
166    _process_selector_ = proc0
167    proc1.running = FALSE
168    proc0.running = TRUE
169  -> State: 1.27 <-
170    proc0.state = exiting
171  -> Input: 1.28 <-
172    _process_selector_ = proc1
173    proc1.running = TRUE
174    proc0.running = FALSE
175  -> State: 1.28 <-
176  -> Input: 1.29 <-
177    _process_selector_ = proc2
178    proc2.running = TRUE
179    proc1.running = FALSE
180  -> State: 1.29 <-
```

```
181     proc2.state = enqueue
182   -> Input: 1.30 <-
183     _process_selector_ = proc3
184     proc3.running = TRUE
185     proc2.running = FALSE
186   -> State: 1.30 <-
187     proc3.state = enqueue
188   -> Input: 1.31 <-
189     _process_selector_ = proc4
190     proc4.running = TRUE
191     proc3.running = FALSE
192   -> State: 1.31 <-
193   -> Input: 1.32 <-
194     _process_selector_ = proc0
195     proc4.running = FALSE
196     proc0.running = TRUE
197   -> State: 1.32 <-
198     semaphore = 2
199     proc0.state = idle
200   -> Input: 1.33 <-
201   -> State: 1.33 <-
202     proc0.state = enqueue
203   -> Input: 1.34 <-
204   -> State: 1.34 <-
205     semaphore = 1
206     apr = 1
207     proc0.state = critical1
208   -> Input: 1.35 <-
209     _process_selector_ = proc1
210     proc1.running = TRUE
211     proc0.running = FALSE
212   -> State: 1.35 <-
213     semaphore = 0
214     apr = 2
215     proc1.state = critical1
216   -> Input: 1.36 <-
217   -> State: 1.36 <-
218     proc1.state = critical2
219   -> Input: 1.37 <-
220   -> State: 1.37 <-
221     proc1.state = exiting
222   -> Input: 1.38 <-
223   -> State: 1.38 <-
224     semaphore = 1
225     proc1.state = idle
226   -> Input: 1.39 <-
227     _process_selector_ = proc2
228     proc2.running = TRUE
229     proc1.running = FALSE
230   -> State: 1.39 <-
231     semaphore = 0
232     apr = 3
233     proc2.state = critical1
234   -> Input: 1.40 <-
235   -> State: 1.40 <-
236     proc2.state = critical2
237   -> Input: 1.41 <-
238   -> State: 1.41 <-
239     proc2.state = exiting
240   -> Input: 1.42 <-
```

```
  -> State: 1.42 <-
    semaphore = 1
    proc2.state = idle
  -> Input: 1.43 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 1.43 <-
    semaphore = 0
    apr = 4
    proc3.state = critical1
  -> Input: 1.44 <-
  -> State: 1.44 <-
    proc3.state = critical2
  -> Input: 1.45 <-
  -> State: 1.45 <-
    proc3.state = exiting
  -> Input: 1.46 <-
  -> State: 1.46 <-
    semaphore = 1
    proc3.state = idle
  -> Input: 1.47 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
  -> State: 1.47 <-
    semaphore = 0
    apr = 0
    proc4.state = critical1
  -> Input: 1.48 <-
  -> State: 1.48 <-
    proc4.state = critical2
  -> Input: 1.49 <-
  -> State: 1.49 <-
    proc4.state = exiting
  -> Input: 1.50 <-
  -> State: 1.50 <-
    semaphore = 1
    proc4.state = idle
  -> Input: 1.51 <-
    _process_selector_ = proc1
    proc4.running = FALSE
    proc1.running = TRUE
  -> State: 1.51 <-
    proc1.state = enqueue
  -> Input: 1.52 <-
    _process_selector_ = proc0
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 1.52 <-
    proc0.state = critical2
  -> Input: 1.53 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc0.running = FALSE
  -> State: 1.53 <-
    proc4.state = enqueue
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = idle) & proc3.state = idle) & proc4.state = idle))  is false
-- as demonstrated by the following execution sequence
```

```
Trace Description: LTL Counterexample
Trace Type: Counterexample
  -> State: 2.1 <-
    semaphore = 2
    apr = 0
    proc0.state = idle
    proc1.state = idle
    proc2.state = idle
    proc3.state = idle
    proc4.state = idle
  -> Input: 2.2 <-
    _process_selector_ = proc0
    running = FALSE
    proc4.running = FALSE
    proc3.running = FALSE
    proc2.running = FALSE
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 2.2 <-
    proc0.state = enqueue
  -> Input: 2.3 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 2.3 <-
    proc1.state = enqueue
  -> Input: 2.4 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 2.4 <-
    proc2.state = enqueue
  -> Input: 2.5 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 2.5 <-
    proc3.state = enqueue
  -> Input: 2.6 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
  -> State: 2.6 <-
    proc4.state = enqueue
  -> Input: 2.7 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 2.7 <-
    semaphore = 1
    apr = 1
    proc0.state = critical1
  -> Input: 2.8 <-
  -> State: 2.8 <-
    proc0.state = critical2
  -> Input: 2.9 <-
  -> State: 2.9 <-
    proc0.state = exiting
  -> Input: 2.10 <-
  -> State: 2.10 <-
```

```
360      semaphore = 2
361      proc0.state = idle
362    -> Input: 2.11 <-
363      _process_selector_ = proc1
364      proc1.running = TRUE
365      proc0.running = FALSE
366    -> State: 2.11 <-
367      semaphore = 1
368      apr = 2
369      proc1.state = critical1
370    -> Input: 2.12 <-
371      _process_selector_ = proc2
372      proc2.running = TRUE
373      proc1.running = FALSE
374    -> State: 2.12 <-
375      semaphore = 0
376      apr = 3
377      proc2.state = critical1
378    -> Input: 2.13 <-
379    -> State: 2.13 <-
380      proc2.state = critical2
381    -> Input: 2.14 <-
382    -> State: 2.14 <-
383      proc2.state = exiting
384    -> Input: 2.15 <-
385    -> State: 2.15 <-
386      semaphore = 1
387      proc2.state = idle
388    -> Input: 2.16 <-
389      _process_selector_ = proc3
390      proc3.running = TRUE
391      proc2.running = FALSE
392    -> State: 2.16 <-
393      semaphore = 0
394      apr = 4
395      proc3.state = critical1
396    -> Input: 2.17 <-
397    -> State: 2.17 <-
398      proc3.state = critical2
399    -> Input: 2.18 <-
400    -> State: 2.18 <-
401      proc3.state = exiting
402    -> Input: 2.19 <-
403    -> State: 2.19 <-
404      semaphore = 1
405      proc3.state = idle
406    -> Input: 2.20 <-
407      _process_selector_ = proc4
408      proc4.running = TRUE
409      proc3.running = FALSE
410    -> State: 2.20 <-
411      semaphore = 0
412      apr = 0
413      proc4.state = critical1
414    -> Input: 2.21 <-
415    -> State: 2.21 <-
416      proc4.state = critical2
417    -> Input: 2.22 <-
418    -> State: 2.22 <-
419      proc4.state = exiting
```

```
  -> Input: 2.23 <-
  -> State: 2.23 <-
    semaphore = 1
    proc4.state = idle
  -> Input: 2.24 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 2.24 <-
    proc0.state = enqueue
  -> Input: 2.25 <-
  -> State: 2.25 <-
    semaphore = 0
    apr = 1
    proc0.state = critical1
  -> Input: 2.26 <-
  -> State: 2.26 <-
    proc0.state = critical2
  -> Input: 2.27 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 2.27 <-
    proc1.state = critical2
  -> Input: 2.28 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc1.running = FALSE
  -> State: 2.28 <-
    proc4.state = enqueue
  -> Input: 2.29 <-
    _process_selector_ = proc2
    proc4.running = FALSE
    proc2.running = TRUE
  -- Loop starts here
  -> State: 2.29 <-
    proc2.state = enqueue
  -> Input: 2.30 <-
    _process_selector_ = proc0
    proc2.running = FALSE
    proc0.running = TRUE
  -> State: 2.30 <-
    proc0.state = exiting
  -> Input: 2.31 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 2.31 <-
    proc1.state = exiting
  -> Input: 2.32 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 2.32 <-
  -> Input: 2.33 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 2.33 <-
    proc3.state = enqueue
```

```
480  -> Input: 2.34 <-
481     _process_selector_ = proc4
482     proc4.running = TRUE
483     proc3.running = FALSE
484  -> State: 2.34 <-
485  -> Input: 2.35 <-
486     _process_selector_ = proc1
487     proc4.running = FALSE
488     proc1.running = TRUE
489  -> State: 2.35 <-
490     semaphore = 1
491     proc1.state = idle
492  -> Input: 2.36 <-
493  -> State: 2.36 <-
494     proc1.state = enqueue
495  -> Input: 2.37 <-
496     _process_selector_ = proc0
497     proc1.running = FALSE
498     proc0.running = TRUE
499  -> State: 2.37 <-
500     semaphore = 2
501     proc0.state = idle
502  -> Input: 2.38 <-
503     _process_selector_ = proc1
504     proc1.running = TRUE
505     proc0.running = FALSE
506  -> State: 2.38 <-
507     semaphore = 1
508     apr = 2
509     proc1.state = critical1
510  -> Input: 2.39 <-
511     _process_selector_ = proc2
512     proc2.running = TRUE
513     proc1.running = FALSE
514  -> State: 2.39 <-
515     semaphore = 0
516     apr = 3
517     proc2.state = critical1
518  -> Input: 2.40 <-
519  -> State: 2.40 <-
520     proc2.state = critical2
521  -> Input: 2.41 <-
522  -> State: 2.41 <-
523     proc2.state = exiting
524  -> Input: 2.42 <-
525  -> State: 2.42 <-
526     semaphore = 1
527     proc2.state = idle
528  -> Input: 2.43 <-
529     _process_selector_ = proc3
530     proc3.running = TRUE
531     proc2.running = FALSE
532  -> State: 2.43 <-
533     semaphore = 0
534     apr = 4
535     proc3.state = critical1
536  -> Input: 2.44 <-
537  -> State: 2.44 <-
538     proc3.state = critical2
539  -> Input: 2.45 <-
```

```
  -> State: 2.45 <-
    proc3.state = exiting
  -> Input: 2.46 <-
  -> State: 2.46 <-
    semaphore = 1
    proc3.state = idle
  -> Input: 2.47 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
  -> State: 2.47 <-
    semaphore = 0
    apr = 0
    proc4.state = critical1
  -> Input: 2.48 <-
  -> State: 2.48 <-
    proc4.state = critical2
  -> Input: 2.49 <-
  -> State: 2.49 <-
    proc4.state = exiting
  -> Input: 2.50 <-
  -> State: 2.50 <-
    semaphore = 1
    proc4.state = idle
  -> Input: 2.51 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 2.51 <-
    proc0.state = enqueue
  -> Input: 2.52 <-
  -> State: 2.52 <-
    semaphore = 0
    apr = 1
    proc0.state = critical1
  -> Input: 2.53 <-
  -> State: 2.53 <-
    proc0.state = critical2
  -> Input: 2.54 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc0.running = FALSE
  -> State: 2.54 <-
    proc2.state = enqueue
  -> Input: 2.55 <-
    _process_selector_ = proc1
    proc2.running = FALSE
    proc1.running = TRUE
  -> State: 2.55 <-
    proc1.state = critical2
  -> Input: 2.56 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc1.running = FALSE
  -> State: 2.56 <-
    proc4.state = enqueue
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = idle) & proc4.state = idle))  is
   ↪ true
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
```

```
    ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state = idle))
    ↪   is true
598 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state =
    ↪ critical2))  is true
```

5.4. Приложение D

Результаты моделирования при инициализации семафора 3-й.

Листинг 5: 3Инициализация семаформа - 3

```
1  *** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:56 2015)
2  *** Enabled addons are: compass
3  *** For more information on NuSMV see <http://nusmv.fbk.eu>
4  *** or email to <nusmv-users@list.fbk.eu>.
5  *** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>
6
7  *** Copyright (c) 2010-2014, Fondazione Bruno Kessler
8
9  *** This version of NuSMV is linked to the CUDD library version 2.4.1
10 *** Copyright (c) 1995-2004, Regents of the University of Colorado
11
12 *** This version of NuSMV is linked to the MiniSat SAT solver.
13 *** See http://minisat.se/MiniSat.html
14 *** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
15 *** Copyright (c) 2007-2010, Niklas Sorensson
16
17 WARNING *** The model contains PROCESSes or ISAs. ***
18 WARNING *** The HRC hierarchy will not be usable. ***
19 -- specification ((((  F proc0.state = critical2 &  F proc1.state = critical2) &
    ↪  F proc2.state = critical2) &  F proc3.state = critical2) &  F proc4.state
    ↪  = critical2)  is true
20 -- specification !( F ((((proc0.state = critical2 & proc1.state = idle) & proc2.
    ↪ state = idle) & proc3.state = idle) & proc4.state = idle))  is false
21 -- as demonstrated by the following execution sequence
22 Trace Description: LTL Counterexample
23 Trace Type: Counterexample
24   -> State: 1.1 <-
25     semaphore = 3
26     apr = 0
27     proc0.state = idle
28     proc1.state = idle
29     proc2.state = idle
30     proc3.state = idle
31     proc4.state = idle
32   -> Input: 1.2 <-
33     _process_selector_ = proc0
34     running = FALSE
35     proc4.running = FALSE
36     proc3.running = FALSE
37     proc2.running = FALSE
38     proc1.running = FALSE
39     proc0.running = TRUE
40   -> State: 1.2 <-
41     proc0.state = enqueue
42   -> Input: 1.3 <-
43     _process_selector_ = proc1
44     proc1.running = TRUE
45     proc0.running = FALSE
```

```
46  -> State: 1.3 <-
47    proc1.state = enqueue
48  -> Input: 1.4 <-
49    _process_selector_ = proc2
50    proc2.running = TRUE
51    proc1.running = FALSE
52  -> State: 1.4 <-
53    proc2.state = enqueue
54  -> Input: 1.5 <-
55    _process_selector_ = proc3
56    proc3.running = TRUE
57    proc2.running = FALSE
58  -> State: 1.5 <-
59    proc3.state = enqueue
60  -> Input: 1.6 <-
61    _process_selector_ = proc4
62    proc4.running = TRUE
63    proc3.running = FALSE
64  -> State: 1.6 <-
65    proc4.state = enqueue
66  -> Input: 1.7 <-
67    _process_selector_ = proc0
68    proc4.running = FALSE
69    proc0.running = TRUE
70  -> State: 1.7 <-
71    semaphore = 2
72    apr = 1
73    proc0.state = critical1
74  -> Input: 1.8 <-
75    _process_selector_ = proc1
76    proc1.running = TRUE
77    proc0.running = FALSE
78  -> State: 1.8 <-
79    semaphore = 1
80    apr = 2
81    proc1.state = critical1
82  -> Input: 1.9 <-
83    _process_selector_ = proc2
84    proc2.running = TRUE
85    proc1.running = FALSE
86  -> State: 1.9 <-
87    semaphore = 0
88    apr = 3
89    proc2.state = critical1
90  -> Input: 1.10 <-
91  -> State: 1.10 <-
92    proc2.state = critical2
93  -> Input: 1.11 <-
94  -> State: 1.11 <-
95    proc2.state = exiting
96  -> Input: 1.12 <-
97  -> State: 1.12 <-
98    semaphore = 1
99    proc2.state = idle
100 -> Input: 1.13 <-
101   _process_selector_ = proc3
102   proc3.running = TRUE
103   proc2.running = FALSE
104 -> State: 1.13 <-
105   semaphore = 0
```

```
106      apr = 4
107      proc3.state = critical1
108    -> Input: 1.14 <-
109    -> State: 1.14 <-
110      proc3.state = critical2
111    -> Input: 1.15 <-
112    -> State: 1.15 <-
113      proc3.state = exiting
114    -> Input: 1.16 <-
115    -> State: 1.16 <-
116      semaphore = 1
117      proc3.state = idle
118    -> Input: 1.17 <-
119      _process_selector_ = proc4
120      proc4.running = TRUE
121      proc3.running = FALSE
122    -> State: 1.17 <-
123      semaphore = 0
124      apr = 0
125      proc4.state = critical1
126    -> Input: 1.18 <-
127    -> State: 1.18 <-
128      proc4.state = critical2
129    -> Input: 1.19 <-
130    -> State: 1.19 <-
131      proc4.state = exiting
132    -> Input: 1.20 <-
133    -> State: 1.20 <-
134      semaphore = 1
135      proc4.state = idle
136    -> Input: 1.21 <-
137      _process_selector_ = proc0
138      proc4.running = FALSE
139      proc0.running = TRUE
140    -> State: 1.21 <-
141      proc0.state = critical2
142    -> Input: 1.22 <-
143      _process_selector_ = proc1
144      proc1.running = TRUE
145      proc0.running = FALSE
146    -> State: 1.22 <-
147      proc1.state = critical2
148    -> Input: 1.23 <-
149    -> State: 1.23 <-
150      proc1.state = exiting
151    -> Input: 1.24 <-
152    -> State: 1.24 <-
153      semaphore = 2
154      proc1.state = idle
155    -> Input: 1.25 <-
156      _process_selector_ = proc4
157      proc4.running = TRUE
158      proc1.running = FALSE
159    -> State: 1.25 <-
160      proc4.state = enqueue
161    -> Input: 1.26 <-
162      _process_selector_ = proc1
163      proc4.running = FALSE
164      proc1.running = TRUE
165    -- Loop starts here
```

```
166  -> State: 1.26 <-
167    proc1.state = enqueue
168  -> Input: 1.27 <-
169    _process_selector_ = proc0
170    proc1.running = FALSE
171    proc0.running = TRUE
172  -> State: 1.27 <-
173    proc0.state = exiting
174  -> Input: 1.28 <-
175    _process_selector_ = proc1
176    proc1.running = TRUE
177    proc0.running = FALSE
178  -> State: 1.28 <-
179  -> Input: 1.29 <-
180    _process_selector_ = proc2
181    proc2.running = TRUE
182    proc1.running = FALSE
183  -> State: 1.29 <-
184    proc2.state = enqueue
185  -> Input: 1.30 <-
186    _process_selector_ = proc3
187    proc3.running = TRUE
188    proc2.running = FALSE
189  -> State: 1.30 <-
190    proc3.state = enqueue
191  -> Input: 1.31 <-
192    _process_selector_ = proc4
193    proc4.running = TRUE
194    proc3.running = FALSE
195  -> State: 1.31 <-
196  -> Input: 1.32 <-
197    _process_selector_ = proc0
198    proc4.running = FALSE
199    proc0.running = TRUE
200  -> State: 1.32 <-
201    semaphore = 3
202    proc0.state = idle
203  -> Input: 1.33 <-
204  -> State: 1.33 <-
205    proc0.state = enqueue
206  -> Input: 1.34 <-
207  -> State: 1.34 <-
208    semaphore = 2
209    apr = 1
210    proc0.state = critical1
211  -> Input: 1.35 <-
212    _process_selector_ = proc1
213    proc1.running = TRUE
214    proc0.running = FALSE
215  -> State: 1.35 <-
216    semaphore = 1
217    apr = 2
218    proc1.state = critical1
219  -> Input: 1.36 <-
220  -> State: 1.36 <-
221    proc1.state = critical2
222  -> Input: 1.37 <-
223  -> State: 1.37 <-
224    proc1.state = exiting
225  -> Input: 1.38 <-
```

```
226  -> State: 1.38 <-
227    semaphore = 2
228    proc1.state = idle
229  -> Input: 1.39 <-
230    _process_selector_ = proc2
231    proc2.running = TRUE
232    proc1.running = FALSE
233  -> State: 1.39 <-
234    semaphore = 1
235    apr = 3
236    proc2.state = critical1
237  -> Input: 1.40 <-
238    _process_selector_ = proc3
239    proc3.running = TRUE
240    proc2.running = FALSE
241  -> State: 1.40 <-
242    semaphore = 0
243    apr = 4
244    proc3.state = critical1
245  -> Input: 1.41 <-
246  -> State: 1.41 <-
247    proc3.state = critical2
248  -> Input: 1.42 <-
249  -> State: 1.42 <-
250    proc3.state = exiting
251  -> Input: 1.43 <-
252  -> State: 1.43 <-
253    semaphore = 1
254    proc3.state = idle
255  -> Input: 1.44 <-
256    _process_selector_ = proc4
257    proc4.running = TRUE
258    proc3.running = FALSE
259  -> State: 1.44 <-
260    semaphore = 0
261    apr = 0
262    proc4.state = critical1
263  -> Input: 1.45 <-
264  -> State: 1.45 <-
265    proc4.state = critical2
266  -> Input: 1.46 <-
267  -> State: 1.46 <-
268    proc4.state = exiting
269  -> Input: 1.47 <-
270  -> State: 1.47 <-
271    semaphore = 1
272    proc4.state = idle
273  -> Input: 1.48 <-
274    _process_selector_ = proc0
275    proc4.running = FALSE
276    proc0.running = TRUE
277  -> State: 1.48 <-
278    proc0.state = critical2
279  -> Input: 1.49 <-
280    _process_selector_ = proc1
281    proc1.running = TRUE
282    proc0.running = FALSE
283  -> State: 1.49 <-
284    proc1.state = enqueue
285  -> Input: 1.50 <-
```

```
286      _process_selector_ = proc2
287      proc2.running = TRUE
288      proc1.running = FALSE
289    -> State: 1.50 <-
290      proc2.state = critical2
291    -> Input: 1.51 <-
292      _process_selector_ = proc4
293      proc4.running = TRUE
294      proc2.running = FALSE
295    -> State: 1.51 <-
296      proc4.state = enqueue
297    -> Input: 1.52 <-
298      _process_selector_ = proc2
299      proc4.running = FALSE
300      proc2.running = TRUE
301    -> State: 1.52 <-
302      proc2.state = exiting
303    -> Input: 1.53 <-
304    -> State: 1.53 <-
305      semaphore = 2
306      proc2.state = idle
307 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
       ↪ proc2.state = idle) & proc3.state = idle) & proc4.state = idle))  is false
308 -- as demonstrated by the following execution sequence
309 Trace Description: LTL Counterexample
310 Trace Type: Counterexample
311    -> State: 2.1 <-
312      semaphore = 3
313      apr = 0
314      proc0.state = idle
315      proc1.state = idle
316      proc2.state = idle
317      proc3.state = idle
318      proc4.state = idle
319    -> Input: 2.2 <-
320      _process_selector_ = proc0
321      running = FALSE
322      proc4.running = FALSE
323      proc3.running = FALSE
324      proc2.running = FALSE
325      proc1.running = FALSE
326      proc0.running = TRUE
327    -> State: 2.2 <-
328      proc0.state = enqueue
329    -> Input: 2.3 <-
330      _process_selector_ = proc1
331      proc1.running = TRUE
332      proc0.running = FALSE
333    -> State: 2.3 <-
334      proc1.state = enqueue
335    -> Input: 2.4 <-
336      _process_selector_ = proc2
337      proc2.running = TRUE
338      proc1.running = FALSE
339    -> State: 2.4 <-
340      proc2.state = enqueue
341    -> Input: 2.5 <-
342      _process_selector_ = proc3
343      proc3.running = TRUE
344      proc2.running = FALSE
```

```
345   -> State: 2.5 <-
346     proc3.state = enqueue
347   -> Input: 2.6 <-
348     _process_selector_ = proc4
349     proc4.running = TRUE
350     proc3.running = FALSE
351   -> State: 2.6 <-
352     proc4.state = enqueue
353   -> Input: 2.7 <-
354     _process_selector_ = proc0
355     proc4.running = FALSE
356     proc0.running = TRUE
357   -> State: 2.7 <-
358     semaphore = 2
359     apr = 1
360     proc0.state = critical1
361   -> Input: 2.8 <-
362     _process_selector_ = proc1
363     proc1.running = TRUE
364     proc0.running = FALSE
365   -> State: 2.8 <-
366     semaphore = 1
367     apr = 2
368     proc1.state = critical1
369   -> Input: 2.9 <-
370     _process_selector_ = proc2
371     proc2.running = TRUE
372     proc1.running = FALSE
373   -> State: 2.9 <-
374     semaphore = 0
375     apr = 3
376     proc2.state = critical1
377   -> Input: 2.10 <-
378   -> State: 2.10 <-
379     proc2.state = critical2
380   -> Input: 2.11 <-
381   -> State: 2.11 <-
382     proc2.state = exiting
383   -> Input: 2.12 <-
384   -> State: 2.12 <-
385     semaphore = 1
386     proc2.state = idle
387   -> Input: 2.13 <-
388     _process_selector_ = proc3
389     proc3.running = TRUE
390     proc2.running = FALSE
391   -> State: 2.13 <-
392     semaphore = 0
393     apr = 4
394     proc3.state = critical1
395   -> Input: 2.14 <-
396   -> State: 2.14 <-
397     proc3.state = critical2
398   -> Input: 2.15 <-
399   -> State: 2.15 <-
400     proc3.state = exiting
401   -> Input: 2.16 <-
402   -> State: 2.16 <-
403     semaphore = 1
404     proc3.state = idle
```

```
405  -> Input: 2.17 <-
406     _process_selector_ = proc4
407     proc4.running = TRUE
408     proc3.running = FALSE
409  -> State: 2.17 <-
410     semaphore = 0
411     apr = 0
412     proc4.state = critical1
413  -> Input: 2.18 <-
414  -> State: 2.18 <-
415     proc4.state = critical2
416  -> Input: 2.19 <-
417  -> State: 2.19 <-
418     proc4.state = exiting
419  -> Input: 2.20 <-
420  -> State: 2.20 <-
421     semaphore = 1
422     proc4.state = idle
423  -> Input: 2.21 <-
424     _process_selector_ = proc0
425     proc4.running = FALSE
426     proc0.running = TRUE
427  -> State: 2.21 <-
428     proc0.state = critical2
429  -> Input: 2.22 <-
430     _process_selector_ = proc1
431     proc1.running = TRUE
432     proc0.running = FALSE
433  -> State: 2.22 <-
434     proc1.state = critical2
435  -> Input: 2.23 <-
436     _process_selector_ = proc4
437     proc4.running = TRUE
438     proc1.running = FALSE
439  -> State: 2.23 <-
440     proc4.state = enqueue
441  -> Input: 2.24 <-
442     _process_selector_ = proc2
443     proc4.running = FALSE
444     proc2.running = TRUE
445  -- Loop starts here
446  -> State: 2.24 <-
447     proc2.state = enqueue
448  -> Input: 2.25 <-
449     _process_selector_ = proc0
450     proc2.running = FALSE
451     proc0.running = TRUE
452  -> State: 2.25 <-
453     proc0.state = exiting
454  -> Input: 2.26 <-
455     _process_selector_ = proc1
456     proc1.running = TRUE
457     proc0.running = FALSE
458  -> State: 2.26 <-
459     proc1.state = exiting
460  -> Input: 2.27 <-
461     _process_selector_ = proc2
462     proc2.running = TRUE
463     proc1.running = FALSE
464  -> State: 2.27 <-
```

```
465  -> Input: 2.28 <-
466     _process_selector_ = proc3
467     proc3.running = TRUE
468     proc2.running = FALSE
469  -> State: 2.28 <-
470     proc3.state = enqueue
471  -> Input: 2.29 <-
472     _process_selector_ = proc4
473     proc4.running = TRUE
474     proc3.running = FALSE
475  -> State: 2.29 <-
476  -> Input: 2.30 <-
477     _process_selector_ = proc1
478     proc4.running = FALSE
479     proc1.running = TRUE
480  -> State: 2.30 <-
481     semaphore = 2
482     proc1.state = idle
483  -> Input: 2.31 <-
484     _process_selector_ = proc0
485     proc1.running = FALSE
486     proc0.running = TRUE
487  -> State: 2.31 <-
488     semaphore = 3
489     proc0.state = idle
490  -> Input: 2.32 <-
491  -> State: 2.32 <-
492     proc0.state = enqueue
493  -> Input: 2.33 <-
494     _process_selector_ = proc1
495     proc1.running = TRUE
496     proc0.running = FALSE
497  -> State: 2.33 <-
498     proc1.state = enqueue
499  -> Input: 2.34 <-
500     _process_selector_ = proc0
501     proc1.running = FALSE
502     proc0.running = TRUE
503  -> State: 2.34 <-
504     semaphore = 2
505     apr = 1
506     proc0.state = critical1
507  -> Input: 2.35 <-
508     _process_selector_ = proc1
509     proc1.running = TRUE
510     proc0.running = FALSE
511  -> State: 2.35 <-
512     semaphore = 1
513     apr = 2
514     proc1.state = critical1
515  -> Input: 2.36 <-
516     _process_selector_ = proc2
517     proc2.running = TRUE
518     proc1.running = FALSE
519  -> State: 2.36 <-
520     semaphore = 0
521     apr = 3
522     proc2.state = critical1
523  -> Input: 2.37 <-
524  -> State: 2.37 <-
```

```
525     proc2.state = critical2
526   -> Input: 2.38 <-
527   -> State: 2.38 <-
528     proc2.state = exiting
529   -> Input: 2.39 <-
530   -> State: 2.39 <-
531     semaphore = 1
532     proc2.state = idle
533   -> Input: 2.40 <-
534     _process_selector_ = proc3
535     proc3.running = TRUE
536     proc2.running = FALSE
537   -> State: 2.40 <-
538     semaphore = 0
539     apr = 4
540     proc3.state = critical1
541   -> Input: 2.41 <-
542   -> State: 2.41 <-
543     proc3.state = critical2
544   -> Input: 2.42 <-
545   -> State: 2.42 <-
546     proc3.state = exiting
547   -> Input: 2.43 <-
548   -> State: 2.43 <-
549     semaphore = 1
550     proc3.state = idle
551   -> Input: 2.44 <-
552     _process_selector_ = proc4
553     proc4.running = TRUE
554     proc3.running = FALSE
555   -> State: 2.44 <-
556     semaphore = 0
557     apr = 0
558     proc4.state = critical1
559   -> Input: 2.45 <-
560   -> State: 2.45 <-
561     proc4.state = critical2
562   -> Input: 2.46 <-
563   -> State: 2.46 <-
564     proc4.state = exiting
565   -> Input: 2.47 <-
566   -> State: 2.47 <-
567     semaphore = 1
568     proc4.state = idle
569   -> Input: 2.48 <-
570     _process_selector_ = proc0
571     proc4.running = FALSE
572     proc0.running = TRUE
573   -> State: 2.48 <-
574     proc0.state = critical2
575   -> Input: 2.49 <-
576     _process_selector_ = proc2
577     proc2.running = TRUE
578     proc0.running = FALSE
579   -> State: 2.49 <-
580     proc2.state = enqueue
581   -> Input: 2.50 <-
582     _process_selector_ = proc1
583     proc2.running = FALSE
584     proc1.running = TRUE
```

```
  -> State: 2.50 <-
    proc1.state = critical2
  -> Input: 2.51 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc1.running = FALSE
  -> State: 2.51 <-
    proc4.state = enqueue
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = idle) & proc4.state = idle))  is
   ↪ false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
Trace Type: Counterexample
  -> State: 3.1 <-
    semaphore = 3
    apr = 0
    proc0.state = idle
    proc1.state = idle
    proc2.state = idle
    proc3.state = idle
    proc4.state = idle
  -> Input: 3.2 <-
    _process_selector_ = proc0
    running = FALSE
    proc4.running = FALSE
    proc3.running = FALSE
    proc2.running = FALSE
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 3.2 <-
    proc0.state = enqueue
  -> Input: 3.3 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 3.3 <-
    proc1.state = enqueue
  -> Input: 3.4 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 3.4 <-
    proc2.state = enqueue
  -> Input: 3.5 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 3.5 <-
    proc3.state = enqueue
  -> Input: 3.6 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
  -> State: 3.6 <-
    proc4.state = enqueue
  -> Input: 3.7 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
```

```
643  -> State: 3.7 <-
644    semaphore = 2
645    apr = 1
646    proc0.state = critical1
647  -> Input: 3.8 <-
648  -> State: 3.8 <-
649    proc0.state = critical2
650  -> Input: 3.9 <-
651  -> State: 3.9 <-
652    proc0.state = exiting
653  -> Input: 3.10 <-
654  -> State: 3.10 <-
655    semaphore = 3
656    proc0.state = idle
657  -> Input: 3.11 <-
658    _process_selector_ = proc1
659    proc1.running = TRUE
660    proc0.running = FALSE
661  -> State: 3.11 <-
662    semaphore = 2
663    apr = 2
664    proc1.state = critical1
665  -> Input: 3.12 <-
666    _process_selector_ = proc2
667    proc2.running = TRUE
668    proc1.running = FALSE
669  -> State: 3.12 <-
670    semaphore = 1
671    apr = 3
672    proc2.state = critical1
673  -> Input: 3.13 <-
674    _process_selector_ = proc3
675    proc3.running = TRUE
676    proc2.running = FALSE
677  -> State: 3.13 <-
678    semaphore = 0
679    apr = 4
680    proc3.state = critical1
681  -> Input: 3.14 <-
682  -> State: 3.14 <-
683    proc3.state = critical2
684  -> Input: 3.15 <-
685  -> State: 3.15 <-
686    proc3.state = exiting
687  -> Input: 3.16 <-
688  -> State: 3.16 <-
689    semaphore = 1
690    proc3.state = idle
691  -> Input: 3.17 <-
692    _process_selector_ = proc4
693    proc4.running = TRUE
694    proc3.running = FALSE
695  -> State: 3.17 <-
696    semaphore = 0
697    apr = 0
698    proc4.state = critical1
699  -> Input: 3.18 <-
700  -> State: 3.18 <-
701    proc4.state = critical2
702  -> Input: 3.19 <-
```

```
  -> State: 3.19 <-
    proc4.state = exiting
  -> Input: 3.20 <-
  -> State: 3.20 <-
    semaphore = 1
    proc4.state = idle
  -> Input: 3.21 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 3.21 <-
    proc0.state = enqueue
  -> Input: 3.22 <-
  -> State: 3.22 <-
    semaphore = 0
    apr = 1
    proc0.state = critical1
  -> Input: 3.23 <-
  -> State: 3.23 <-
    proc0.state = critical2
  -> Input: 3.24 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 3.24 <-
    proc1.state = critical2
  -> Input: 3.25 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 3.25 <-
    proc2.state = critical2
  -> Input: 3.26 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc2.running = FALSE
  -> State: 3.26 <-
    proc4.state = enqueue
  -> Input: 3.27 <-
    _process_selector_ = proc3
    proc4.running = FALSE
    proc3.running = TRUE
  -- Loop starts here
  -> State: 3.27 <-
    proc3.state = enqueue
  -> Input: 3.28 <-
    _process_selector_ = proc0
    proc3.running = FALSE
    proc0.running = TRUE
  -> State: 3.28 <-
    proc0.state = exiting
  -> Input: 3.29 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 3.29 <-
    proc1.state = exiting
  -> Input: 3.30 <-
    _process_selector_ = proc2
    proc2.running = TRUE
```

```
763      proc1.running = FALSE
764    -> State: 3.30 <-
765      proc2.state = exiting
766    -> Input: 3.31 <-
767      _process_selector_ = proc3
768      proc3.running = TRUE
769      proc2.running = FALSE
770    -> State: 3.31 <-
771    -> Input: 3.32 <-
772      _process_selector_ = proc4
773      proc4.running = TRUE
774      proc3.running = FALSE
775    -> State: 3.32 <-
776    -> Input: 3.33 <-
777      _process_selector_ = proc2
778      proc4.running = FALSE
779      proc2.running = TRUE
780    -> State: 3.33 <-
781      semaphore = 1
782      proc2.state = idle
783    -> Input: 3.34 <-
784      _process_selector_ = proc1
785      proc2.running = FALSE
786      proc1.running = TRUE
787    -> State: 3.34 <-
788      semaphore = 2
789      proc1.state = idle
790    -> Input: 3.35 <-
791      _process_selector_ = proc0
792      proc1.running = FALSE
793      proc0.running = TRUE
794    -> State: 3.35 <-
795      semaphore = 3
796      proc0.state = idle
797    -> Input: 3.36 <-
798      _process_selector_ = proc1
799      proc1.running = TRUE
800      proc0.running = FALSE
801    -> State: 3.36 <-
802      proc1.state = enqueue
803    -> Input: 3.37 <-
804      _process_selector_ = proc2
805      proc2.running = TRUE
806      proc1.running = FALSE
807    -> State: 3.37 <-
808      proc2.state = enqueue
809    -> Input: 3.38 <-
810      _process_selector_ = proc1
811      proc2.running = FALSE
812      proc1.running = TRUE
813    -> State: 3.38 <-
814      semaphore = 2
815      apr = 2
816      proc1.state = critical1
817    -> Input: 3.39 <-
818      _process_selector_ = proc2
819      proc2.running = TRUE
820      proc1.running = FALSE
821    -> State: 3.39 <-
822      semaphore = 1
```

```
823     apr = 3
824     proc2.state = critical1
825   -> Input: 3.40 <-
826     _process_selector_ = proc3
827     proc3.running = TRUE
828     proc2.running = FALSE
829   -> State: 3.40 <-
830     semaphore = 0
831     apr = 4
832     proc3.state = critical1
833   -> Input: 3.41 <-
834   -> State: 3.41 <-
835     proc3.state = critical2
836   -> Input: 3.42 <-
837   -> State: 3.42 <-
838     proc3.state = exiting
839   -> Input: 3.43 <-
840   -> State: 3.43 <-
841     semaphore = 1
842     proc3.state = idle
843   -> Input: 3.44 <-
844     _process_selector_ = proc4
845     proc4.running = TRUE
846     proc3.running = FALSE
847   -> State: 3.44 <-
848     semaphore = 0
849     apr = 0
850     proc4.state = critical1
851   -> Input: 3.45 <-
852   -> State: 3.45 <-
853     proc4.state = critical2
854   -> Input: 3.46 <-
855   -> State: 3.46 <-
856     proc4.state = exiting
857   -> Input: 3.47 <-
858   -> State: 3.47 <-
859     semaphore = 1
860     proc4.state = idle
861   -> Input: 3.48 <-
862     _process_selector_ = proc0
863     proc4.running = FALSE
864     proc0.running = TRUE
865   -> State: 3.48 <-
866     proc0.state = enqueue
867   -> Input: 3.49 <-
868   -> State: 3.49 <-
869     semaphore = 0
870     apr = 1
871     proc0.state = critical1
872   -> Input: 3.50 <-
873   -> State: 3.50 <-
874     proc0.state = critical2
875   -> Input: 3.51 <-
876     _process_selector_ = proc1
877     proc1.running = TRUE
878     proc0.running = FALSE
879   -> State: 3.51 <-
880     proc1.state = critical2
881   -> Input: 3.52 <-
882     _process_selector_ = proc3
```

```
883     proc3.running = TRUE
884     proc1.running = FALSE
885   -> State: 3.52 <-
886     proc3.state = enqueue
887   -> Input: 3.53 <-
888     _process_selector_ = proc2
889     proc3.running = FALSE
890     proc2.running = TRUE
891   -> State: 3.53 <-
892     proc2.state = critical2
893   -> Input: 3.54 <-
894     _process_selector_ = proc4
895     proc4.running = TRUE
896     proc2.running = FALSE
897   -> State: 3.54 <-
898     proc4.state = enqueue
899 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state = idle))
    ↪   is true
900 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state =
    ↪ critical2))  is true
```

5.5. Приложение E

Результаты моделирования при инициализации семафора 4-й

Листинг 6: Инициализация семаформа - 4

```
1  *** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:56 2015)
2  *** Enabled addons are: compass
3  *** For more information on NuSMV see <http://nusmv.fbk.eu>
4  *** or email to <nusmv-users@list.fbk.eu>.
5  *** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>
6
7  *** Copyright (c) 2010-2014, Fondazione Bruno Kessler
8
9  *** This version of NuSMV is linked to the CUDD library version 2.4.1
10 *** Copyright (c) 1995-2004, Regents of the University of Colorado
11
12 *** This version of NuSMV is linked to the MiniSat SAT solver.
13 *** See http://minisat.se/MiniSat.html
14 *** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
15 *** Copyright (c) 2007-2010, Niklas Sorensson
16
17 WARNING *** The model contains PROCESSes or ISAs. ***
18 WARNING *** The HRC hierarchy will not be usable. ***
19 -- specification ((((  F proc0.state = critical2 &  F proc1.state = critical2) &
    ↪  F proc2.state = critical2) &  F proc3.state = critical2) &  F proc4.state
    ↪  = critical2)  is true
20 -- specification !( F ((((proc0.state = critical2 & proc1.state = idle) & proc2.
    ↪ state = idle) & proc3.state = idle) & proc4.state = idle))  is false
21 -- as demonstrated by the following execution sequence
22 Trace Description: LTL Counterexample
23 Trace Type: Counterexample
24   -> State: 1.1 <-
25     semaphore = 4
26     apr = 0
27     proc0.state = idle
28     proc1.state = idle
```

```
29      proc2.state = idle
30      proc3.state = idle
31      proc4.state = idle
32    -> Input: 1.2 <-
33      _process_selector_ = proc0
34      running = FALSE
35      proc4.running = FALSE
36      proc3.running = FALSE
37      proc2.running = FALSE
38      proc1.running = FALSE
39      proc0.running = TRUE
40    -> State: 1.2 <-
41      proc0.state = enqueue
42    -> Input: 1.3 <-
43      _process_selector_ = proc1
44      proc1.running = TRUE
45      proc0.running = FALSE
46    -> State: 1.3 <-
47      proc1.state = enqueue
48    -> Input: 1.4 <-
49      _process_selector_ = proc2
50      proc2.running = TRUE
51      proc1.running = FALSE
52    -> State: 1.4 <-
53      proc2.state = enqueue
54    -> Input: 1.5 <-
55      _process_selector_ = proc3
56      proc3.running = TRUE
57      proc2.running = FALSE
58    -> State: 1.5 <-
59      proc3.state = enqueue
60    -> Input: 1.6 <-
61      _process_selector_ = proc4
62      proc4.running = TRUE
63      proc3.running = FALSE
64    -> State: 1.6 <-
65      proc4.state = enqueue
66    -> Input: 1.7 <-
67      _process_selector_ = proc0
68      proc4.running = FALSE
69      proc0.running = TRUE
70    -> State: 1.7 <-
71      semaphore = 3
72      apr = 1
73      proc0.state = critical1
74    -> Input: 1.8 <-
75      _process_selector_ = proc1
76      proc1.running = TRUE
77      proc0.running = FALSE
78    -> State: 1.8 <-
79      semaphore = 2
80      apr = 2
81      proc1.state = critical1
82    -> Input: 1.9 <-
83      _process_selector_ = proc2
84      proc2.running = TRUE
85      proc1.running = FALSE
86    -> State: 1.9 <-
87      semaphore = 1
88      apr = 3
```

```
89      proc2.state = critical1
90    -> Input: 1.10 <-
91      _process_selector_ = proc3
92      proc3.running = TRUE
93      proc2.running = FALSE
94    -> State: 1.10 <-
95      semaphore = 0
96      apr = 4
97      proc3.state = critical1
98    -> Input: 1.11 <-
99    -> State: 1.11 <-
100     proc3.state = critical2
101   -> Input: 1.12 <-
102   -> State: 1.12 <-
103     proc3.state = exiting
104   -> Input: 1.13 <-
105   -> State: 1.13 <-
106     semaphore = 1
107     proc3.state = idle
108   -> Input: 1.14 <-
109     _process_selector_ = proc4
110     proc4.running = TRUE
111     proc3.running = FALSE
112   -> State: 1.14 <-
113     semaphore = 0
114     apr = 0
115     proc4.state = critical1
116   -> Input: 1.15 <-
117   -> State: 1.15 <-
118     proc4.state = critical2
119   -> Input: 1.16 <-
120   -> State: 1.16 <-
121     proc4.state = exiting
122   -> Input: 1.17 <-
123   -> State: 1.17 <-
124     semaphore = 1
125     proc4.state = idle
126   -> Input: 1.18 <-
127     _process_selector_ = proc1
128     proc4.running = FALSE
129     proc1.running = TRUE
130   -> State: 1.18 <-
131     proc1.state = critical2
132   -> Input: 1.19 <-
133     _process_selector_ = proc2
134     proc2.running = TRUE
135     proc1.running = FALSE
136   -> State: 1.19 <-
137     proc2.state = critical2
138   -> Input: 1.20 <-
139   -> State: 1.20 <-
140     proc2.state = exiting
141   -> Input: 1.21 <-
142     _process_selector_ = proc1
143     proc2.running = FALSE
144     proc1.running = TRUE
145   -> State: 1.21 <-
146     proc1.state = exiting
147   -> Input: 1.22 <-
148     _process_selector_ = proc2
```

```
149      proc2.running = TRUE
150      proc1.running = FALSE
151    -> State: 1.22 <-
152      semaphore = 2
153      proc2.state = idle
154    -> Input: 1.23 <-
155      _process_selector_ = proc1
156      proc2.running = FALSE
157      proc1.running = TRUE
158    -> State: 1.23 <-
159      semaphore = 3
160      proc1.state = idle
161    -> Input: 1.24 <-
162      _process_selector_ = proc0
163      proc1.running = FALSE
164      proc0.running = TRUE
165    -> State: 1.24 <-
166      proc0.state = critical2
167    -> Input: 1.25 <-
168      _process_selector_ = proc4
169      proc4.running = TRUE
170      proc0.running = FALSE
171    -> State: 1.25 <-
172      proc4.state = enqueue
173    -> Input: 1.26 <-
174      _process_selector_ = proc1
175      proc4.running = FALSE
176      proc1.running = TRUE
177    -- Loop starts here
178    -> State: 1.26 <-
179      proc1.state = enqueue
180    -> Input: 1.27 <-
181      _process_selector_ = proc0
182      proc1.running = FALSE
183      proc0.running = TRUE
184    -> State: 1.27 <-
185      proc0.state = exiting
186    -> Input: 1.28 <-
187      _process_selector_ = proc1
188      proc1.running = TRUE
189      proc0.running = FALSE
190    -> State: 1.28 <-
191    -> Input: 1.29 <-
192      _process_selector_ = proc2
193      proc2.running = TRUE
194      proc1.running = FALSE
195    -> State: 1.29 <-
196      proc2.state = enqueue
197    -> Input: 1.30 <-
198      _process_selector_ = proc3
199      proc3.running = TRUE
200      proc2.running = FALSE
201    -> State: 1.30 <-
202      proc3.state = enqueue
203    -> Input: 1.31 <-
204      _process_selector_ = proc4
205      proc4.running = TRUE
206      proc3.running = FALSE
207    -> State: 1.31 <-
208    -> Input: 1.32 <-
```

```
209      _process_selector_ = proc0
210      proc4.running = FALSE
211      proc0.running = TRUE
212    -> State: 1.32 <-
213      semaphore = 4
214      proc0.state = idle
215    -> Input: 1.33 <-
216    -> State: 1.33 <-
217      proc0.state = enqueue
218    -> Input: 1.34 <-
219    -> State: 1.34 <-
220      semaphore = 3
221      apr = 1
222      proc0.state = critical1
223    -> Input: 1.35 <-
224      _process_selector_ = proc1
225      proc1.running = TRUE
226      proc0.running = FALSE
227    -> State: 1.35 <-
228      semaphore = 2
229      apr = 2
230      proc1.state = critical1
231    -> Input: 1.36 <-
232      _process_selector_ = proc2
233      proc2.running = TRUE
234      proc1.running = FALSE
235    -> State: 1.36 <-
236      semaphore = 1
237      apr = 3
238      proc2.state = critical1
239    -> Input: 1.37 <-
240      _process_selector_ = proc3
241      proc3.running = TRUE
242      proc2.running = FALSE
243    -> State: 1.37 <-
244      semaphore = 0
245      apr = 4
246      proc3.state = critical1
247    -> Input: 1.38 <-
248    -> State: 1.38 <-
249      proc3.state = critical2
250    -> Input: 1.39 <-
251    -> State: 1.39 <-
252      proc3.state = exiting
253    -> Input: 1.40 <-
254    -> State: 1.40 <-
255      semaphore = 1
256      proc3.state = idle
257    -> Input: 1.41 <-
258      _process_selector_ = proc4
259      proc4.running = TRUE
260      proc3.running = FALSE
261    -> State: 1.41 <-
262      semaphore = 0
263      apr = 0
264      proc4.state = critical1
265    -> Input: 1.42 <-
266    -> State: 1.42 <-
267      proc4.state = critical2
268    -> Input: 1.43 <-
```

```
  -> State: 1.43 <-
    proc4.state = exiting
  -> Input: 1.44 <-
  -> State: 1.44 <-
    semaphore = 1
    proc4.state = idle
  -> Input: 1.45 <-
    _process_selector_ = proc1
    proc4.running = FALSE
    proc1.running = TRUE
  -> State: 1.45 <-
    proc1.state = critical2
  -> Input: 1.46 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 1.46 <-
    proc2.state = critical2
  -> Input: 1.47 <-
  -> State: 1.47 <-
    proc2.state = exiting
  -> Input: 1.48 <-
    _process_selector_ = proc1
    proc2.running = FALSE
    proc1.running = TRUE
  -> State: 1.48 <-
    proc1.state = exiting
  -> Input: 1.49 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 1.49 <-
    semaphore = 2
    proc2.state = idle
  -> Input: 1.50 <-
    _process_selector_ = proc1
    proc2.running = FALSE
    proc1.running = TRUE
  -> State: 1.50 <-
    semaphore = 3
    proc1.state = idle
  -> Input: 1.51 <-
  -> State: 1.51 <-
    proc1.state = enqueue
  -> Input: 1.52 <-
    _process_selector_ = proc0
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 1.52 <-
    proc0.state = critical2
  -> Input: 1.53 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc0.running = FALSE
  -> State: 1.53 <-
    proc4.state = enqueue
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = idle) & proc3.state = idle) & proc4.state = idle))  is false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
```

```
Trace Type: Counterexample
  -> State: 2.1 <-
    semaphore = 4
    apr = 0
    proc0.state = idle
    proc1.state = idle
    proc2.state = idle
    proc3.state = idle
    proc4.state = idle
  -> Input: 2.2 <-
    _process_selector_ = proc0
    running = FALSE
    proc4.running = FALSE
    proc3.running = FALSE
    proc2.running = FALSE
    proc1.running = FALSE
    proc0.running = TRUE
  -> State: 2.2 <-
    proc0.state = enqueue
  -> Input: 2.3 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 2.3 <-
    proc1.state = enqueue
  -> Input: 2.4 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 2.4 <-
    proc2.state = enqueue
  -> Input: 2.5 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 2.5 <-
    proc3.state = enqueue
  -> Input: 2.6 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
  -> State: 2.6 <-
    proc4.state = enqueue
  -> Input: 2.7 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 2.7 <-
    semaphore = 3
    apr = 1
    proc0.state = critical1
  -> Input: 2.8 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 2.8 <-
    semaphore = 2
    apr = 2
    proc1.state = critical1
  -> Input: 2.9 <-
```

```
388      _process_selector_ = proc2
389      proc2.running = TRUE
390      proc1.running = FALSE
391    -> State: 2.9 <-
392      semaphore = 1
393      apr = 3
394      proc2.state = critical1
395    -> Input: 2.10 <-
396      _process_selector_ = proc3
397      proc3.running = TRUE
398      proc2.running = FALSE
399    -> State: 2.10 <-
400      semaphore = 0
401      apr = 4
402      proc3.state = critical1
403    -> Input: 2.11 <-
404    -> State: 2.11 <-
405      proc3.state = critical2
406    -> Input: 2.12 <-
407    -> State: 2.12 <-
408      proc3.state = exiting
409    -> Input: 2.13 <-
410    -> State: 2.13 <-
411      semaphore = 1
412      proc3.state = idle
413    -> Input: 2.14 <-
414      _process_selector_ = proc4
415      proc4.running = TRUE
416      proc3.running = FALSE
417    -> State: 2.14 <-
418      semaphore = 0
419      apr = 0
420      proc4.state = critical1
421    -> Input: 2.15 <-
422    -> State: 2.15 <-
423      proc4.state = critical2
424    -> Input: 2.16 <-
425    -> State: 2.16 <-
426      proc4.state = exiting
427    -> Input: 2.17 <-
428    -> State: 2.17 <-
429      semaphore = 1
430      proc4.state = idle
431    -> Input: 2.18 <-
432      _process_selector_ = proc0
433      proc4.running = FALSE
434      proc0.running = TRUE
435    -> State: 2.18 <-
436      proc0.state = critical2
437    -> Input: 2.19 <-
438      _process_selector_ = proc1
439      proc1.running = TRUE
440      proc0.running = FALSE
441    -> State: 2.19 <-
442      proc1.state = critical2
443    -> Input: 2.20 <-
444      _process_selector_ = proc2
445      proc2.running = TRUE
446      proc1.running = FALSE
447    -> State: 2.20 <-
```

```
448     proc2.state = critical2
449   -> Input: 2.21 <-
450   -> State: 2.21 <-
451     proc2.state = exiting
452   -> Input: 2.22 <-
453   -> State: 2.22 <-
454     semaphore = 2
455     proc2.state = idle
456   -> Input: 2.23 <-
457     _process_selector_ = proc4
458     proc4.running = TRUE
459     proc2.running = FALSE
460   -> State: 2.23 <-
461     proc4.state = enqueue
462   -> Input: 2.24 <-
463     _process_selector_ = proc2
464     proc4.running = FALSE
465     proc2.running = TRUE
466   -- Loop starts here
467   -> State: 2.24 <-
468     proc2.state = enqueue
469   -> Input: 2.25 <-
470     _process_selector_ = proc0
471     proc2.running = FALSE
472     proc0.running = TRUE
473   -> State: 2.25 <-
474     proc0.state = exiting
475   -> Input: 2.26 <-
476     _process_selector_ = proc1
477     proc1.running = TRUE
478     proc0.running = FALSE
479   -> State: 2.26 <-
480     proc1.state = exiting
481   -> Input: 2.27 <-
482     _process_selector_ = proc2
483     proc2.running = TRUE
484     proc1.running = FALSE
485   -> State: 2.27 <-
486   -> Input: 2.28 <-
487     _process_selector_ = proc3
488     proc3.running = TRUE
489     proc2.running = FALSE
490   -> State: 2.28 <-
491     proc3.state = enqueue
492   -> Input: 2.29 <-
493     _process_selector_ = proc4
494     proc4.running = TRUE
495     proc3.running = FALSE
496   -> State: 2.29 <-
497   -> Input: 2.30 <-
498     _process_selector_ = proc0
499     proc4.running = FALSE
500     proc0.running = TRUE
501   -> State: 2.30 <-
502     semaphore = 3
503     proc0.state = idle
504   -> Input: 2.31 <-
505   -> State: 2.31 <-
506     proc0.state = enqueue
507   -> Input: 2.32 <-
```

```
-> State: 2.32 <-
  semaphore = 2
  apr = 1
  proc0.state = critical1
-> Input: 2.33 <-
  _process_selector_ = proc1
  proc1.running = TRUE
  proc0.running = FALSE
-> State: 2.33 <-
  semaphore = 3
  proc1.state = idle
-> Input: 2.34 <-
-> State: 2.34 <-
  proc1.state = enqueue
-> Input: 2.35 <-
-> State: 2.35 <-
  semaphore = 2
  apr = 2
  proc1.state = critical1
-> Input: 2.36 <-
  _process_selector_ = proc2
  proc2.running = TRUE
  proc1.running = FALSE
-> State: 2.36 <-
  semaphore = 1
  apr = 3
  proc2.state = critical1
-> Input: 2.37 <-
-> State: 2.37 <-
  proc2.state = critical2
-> Input: 2.38 <-
-> State: 2.38 <-
  proc2.state = exiting
-> Input: 2.39 <-
-> State: 2.39 <-
  semaphore = 2
  proc2.state = idle
-> Input: 2.40 <-
  _process_selector_ = proc3
  proc3.running = TRUE
  proc2.running = FALSE
-> State: 2.40 <-
  semaphore = 1
  apr = 4
  proc3.state = critical1
-> Input: 2.41 <-
  _process_selector_ = proc4
  proc4.running = TRUE
  proc3.running = FALSE
-> State: 2.41 <-
  semaphore = 0
  apr = 0
  proc4.state = critical1
-> Input: 2.42 <-
-> State: 2.42 <-
  proc4.state = critical2
-> Input: 2.43 <-
-> State: 2.43 <-
  proc4.state = exiting
-> Input: 2.44 <-
```

```
  -> State: 2.44 <-
    semaphore = 1
    proc4.state = idle
  -> Input: 2.45 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
  -> State: 2.45 <-
    proc0.state = critical2
  -> Input: 2.46 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 2.46 <-
    proc1.state = critical2
  -> Input: 2.47 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
  -> State: 2.47 <-
    proc2.state = enqueue
  -> Input: 2.48 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 2.48 <-
    proc3.state = critical2
  -> Input: 2.49 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
  -> State: 2.49 <-
    proc4.state = enqueue
  -> Input: 2.50 <-
    _process_selector_ = proc3
    proc4.running = FALSE
    proc3.running = TRUE
  -> State: 2.50 <-
    proc3.state = exiting
  -> Input: 2.51 <-
  -> State: 2.51 <-
    semaphore = 2
    proc3.state = idle
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = idle) & proc4.state = idle))  is
   ↪ false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
Trace Type: Counterexample
  -> State: 3.1 <-
    semaphore = 4
    apr = 0
    proc0.state = idle
    proc1.state = idle
    proc2.state = idle
    proc3.state = idle
    proc4.state = idle
  -> Input: 3.2 <-
    _process_selector_ = proc0
    running = FALSE
```

```
626      proc4.running = FALSE
627      proc3.running = FALSE
628      proc2.running = FALSE
629      proc1.running = FALSE
630      proc0.running = TRUE
631    -> State: 3.2 <-
632      proc0.state = enqueue
633    -> Input: 3.3 <-
634      _process_selector_ = proc1
635      proc1.running = TRUE
636      proc0.running = FALSE
637    -> State: 3.3 <-
638      proc1.state = enqueue
639    -> Input: 3.4 <-
640      _process_selector_ = proc2
641      proc2.running = TRUE
642      proc1.running = FALSE
643    -> State: 3.4 <-
644      proc2.state = enqueue
645    -> Input: 3.5 <-
646      _process_selector_ = proc3
647      proc3.running = TRUE
648      proc2.running = FALSE
649    -> State: 3.5 <-
650      proc3.state = enqueue
651    -> Input: 3.6 <-
652      _process_selector_ = proc4
653      proc4.running = TRUE
654      proc3.running = FALSE
655    -> State: 3.6 <-
656      proc4.state = enqueue
657    -> Input: 3.7 <-
658      _process_selector_ = proc0
659      proc4.running = FALSE
660      proc0.running = TRUE
661    -> State: 3.7 <-
662      semaphore = 3
663      apr = 1
664      proc0.state = critical1
665    -> Input: 3.8 <-
666      _process_selector_ = proc1
667      proc1.running = TRUE
668      proc0.running = FALSE
669    -> State: 3.8 <-
670      semaphore = 2
671      apr = 2
672      proc1.state = critical1
673    -> Input: 3.9 <-
674      _process_selector_ = proc2
675      proc2.running = TRUE
676      proc1.running = FALSE
677    -> State: 3.9 <-
678      semaphore = 1
679      apr = 3
680      proc2.state = critical1
681    -> Input: 3.10 <-
682      _process_selector_ = proc3
683      proc3.running = TRUE
684      proc2.running = FALSE
685    -> State: 3.10 <-
```

```
686      semaphore = 0
687      apr = 4
688      proc3.state = critical1
689    -> Input: 3.11 <-
690    -> State: 3.11 <-
691      proc3.state = critical2
692    -> Input: 3.12 <-
693    -> State: 3.12 <-
694      proc3.state = exiting
695    -> Input: 3.13 <-
696    -> State: 3.13 <-
697      semaphore = 1
698      proc3.state = idle
699    -> Input: 3.14 <-
700      _process_selector_ = proc4
701      proc4.running = TRUE
702      proc3.running = FALSE
703    -> State: 3.14 <-
704      semaphore = 0
705      apr = 0
706      proc4.state = critical1
707    -> Input: 3.15 <-
708    -> State: 3.15 <-
709      proc4.state = critical2
710    -> Input: 3.16 <-
711    -> State: 3.16 <-
712      proc4.state = exiting
713    -> Input: 3.17 <-
714    -> State: 3.17 <-
715      semaphore = 1
716      proc4.state = idle
717    -> Input: 3.18 <-
718      _process_selector_ = proc0
719      proc4.running = FALSE
720      proc0.running = TRUE
721    -> State: 3.18 <-
722      proc0.state = critical2
723    -> Input: 3.19 <-
724      _process_selector_ = proc1
725      proc1.running = TRUE
726      proc0.running = FALSE
727    -> State: 3.19 <-
728      proc1.state = critical2
729    -> Input: 3.20 <-
730      _process_selector_ = proc2
731      proc2.running = TRUE
732      proc1.running = FALSE
733    -> State: 3.20 <-
734      proc2.state = critical2
735    -> Input: 3.21 <-
736      _process_selector_ = proc4
737      proc4.running = TRUE
738      proc2.running = FALSE
739    -> State: 3.21 <-
740      proc4.state = enqueue
741    -> Input: 3.22 <-
742      _process_selector_ = proc3
743      proc4.running = FALSE
744      proc3.running = TRUE
745    -- Loop starts here
```

```
746  -> State: 3.22 <-
747    proc3.state = enqueue
748  -> Input: 3.23 <-
749    _process_selector_ = proc0
750    proc3.running = FALSE
751    proc0.running = TRUE
752  -> State: 3.23 <-
753    proc0.state = exiting
754  -> Input: 3.24 <-
755    _process_selector_ = proc1
756    proc1.running = TRUE
757    proc0.running = FALSE
758  -> State: 3.24 <-
759    proc1.state = exiting
760  -> Input: 3.25 <-
761    _process_selector_ = proc2
762    proc2.running = TRUE
763    proc1.running = FALSE
764  -> State: 3.25 <-
765    proc2.state = exiting
766  -> Input: 3.26 <-
767    _process_selector_ = proc3
768    proc3.running = TRUE
769    proc2.running = FALSE
770  -> State: 3.26 <-
771  -> Input: 3.27 <-
772    _process_selector_ = proc4
773    proc4.running = TRUE
774    proc3.running = FALSE
775  -> State: 3.27 <-
776  -> Input: 3.28 <-
777    _process_selector_ = proc2
778    proc4.running = FALSE
779    proc2.running = TRUE
780  -> State: 3.28 <-
781    semaphore = 2
782    proc2.state = idle
783  -> Input: 3.29 <-
784    _process_selector_ = proc0
785    proc2.running = FALSE
786    proc0.running = TRUE
787  -> State: 3.29 <-
788    semaphore = 3
789    proc0.state = idle
790  -> Input: 3.30 <-
791  -> State: 3.30 <-
792    proc0.state = enqueue
793  -> Input: 3.31 <-
794  -> State: 3.31 <-
795    semaphore = 2
796    apr = 1
797    proc0.state = critical1
798  -> Input: 3.32 <-
799    _process_selector_ = proc1
800    proc1.running = TRUE
801    proc0.running = FALSE
802  -> State: 3.32 <-
803    semaphore = 3
804    proc1.state = idle
805  -> Input: 3.33 <-
```

```
806  -> State: 3.33 <-
807    proc1.state = enqueue
808  -> Input: 3.34 <-
809    _process_selector_ = proc2
810    proc2.running = TRUE
811    proc1.running = FALSE
812  -> State: 3.34 <-
813    proc2.state = enqueue
814  -> Input: 3.35 <-
815    _process_selector_ = proc1
816    proc2.running = FALSE
817    proc1.running = TRUE
818  -> State: 3.35 <-
819    semaphore = 2
820    apr = 2
821    proc1.state = critical1
822  -> Input: 3.36 <-
823    _process_selector_ = proc2
824    proc2.running = TRUE
825    proc1.running = FALSE
826  -> State: 3.36 <-
827    semaphore = 1
828    apr = 3
829    proc2.state = critical1
830  -> Input: 3.37 <-
831    _process_selector_ = proc3
832    proc3.running = TRUE
833    proc2.running = FALSE
834  -> State: 3.37 <-
835    semaphore = 0
836    apr = 4
837    proc3.state = critical1
838  -> Input: 3.38 <-
839  -> State: 3.38 <-
840    proc3.state = critical2
841  -> Input: 3.39 <-
842  -> State: 3.39 <-
843    proc3.state = exiting
844  -> Input: 3.40 <-
845  -> State: 3.40 <-
846    semaphore = 1
847    proc3.state = idle
848  -> Input: 3.41 <-
849    _process_selector_ = proc4
850    proc4.running = TRUE
851    proc3.running = FALSE
852  -> State: 3.41 <-
853    semaphore = 0
854    apr = 0
855    proc4.state = critical1
856  -> Input: 3.42 <-
857  -> State: 3.42 <-
858    proc4.state = critical2
859  -> Input: 3.43 <-
860  -> State: 3.43 <-
861    proc4.state = exiting
862  -> Input: 3.44 <-
863  -> State: 3.44 <-
864    semaphore = 1
865    proc4.state = idle
```

```
  -> Input: 3.45 <-
     _process_selector_ = proc0
     proc4.running = FALSE
     proc0.running = TRUE
  -> State: 3.45 <-
     proc0.state = critical2
  -> Input: 3.46 <-
     _process_selector_ = proc1
     proc1.running = TRUE
     proc0.running = FALSE
  -> State: 3.46 <-
     proc1.state = critical2
  -> Input: 3.47 <-
     _process_selector_ = proc3
     proc3.running = TRUE
     proc1.running = FALSE
  -> State: 3.47 <-
     proc3.state = enqueue
  -> Input: 3.48 <-
     _process_selector_ = proc2
     proc3.running = FALSE
     proc2.running = TRUE
  -> State: 3.48 <-
     proc2.state = critical2
  -> Input: 3.49 <-
     _process_selector_ = proc4
     proc4.running = TRUE
     proc2.running = FALSE
  -> State: 3.49 <-
     proc4.state = enqueue
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state = idle))
   ↪    is false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
Trace Type: Counterexample
  -> State: 4.1 <-
     semaphore = 4
     apr = 0
     proc0.state = idle
     proc1.state = idle
     proc2.state = idle
     proc3.state = idle
     proc4.state = idle
  -> Input: 4.2 <-
     _process_selector_ = proc0
     running = FALSE
     proc4.running = FALSE
     proc3.running = FALSE
     proc2.running = FALSE
     proc1.running = FALSE
     proc0.running = TRUE
  -> State: 4.2 <-
     proc0.state = enqueue
  -> Input: 4.3 <-
     _process_selector_ = proc1
     proc1.running = TRUE
     proc0.running = FALSE
  -> State: 4.3 <-
     proc1.state = enqueue
```

```
924  -> Input: 4.4 <-
925     _process_selector_ = proc2
926     proc2.running = TRUE
927     proc1.running = FALSE
928  -> State: 4.4 <-
929     proc2.state = enqueue
930  -> Input: 4.5 <-
931     _process_selector_ = proc3
932     proc3.running = TRUE
933     proc2.running = FALSE
934  -> State: 4.5 <-
935     proc3.state = enqueue
936  -> Input: 4.6 <-
937     _process_selector_ = proc4
938     proc4.running = TRUE
939     proc3.running = FALSE
940  -> State: 4.6 <-
941     proc4.state = enqueue
942  -> Input: 4.7 <-
943     _process_selector_ = proc0
944     proc4.running = FALSE
945     proc0.running = TRUE
946  -> State: 4.7 <-
947     semaphore = 3
948     apr = 1
949     proc0.state = critical1
950  -> Input: 4.8 <-
951  -> State: 4.8 <-
952     proc0.state = critical2
953  -> Input: 4.9 <-
954  -> State: 4.9 <-
955     proc0.state = exiting
956  -> Input: 4.10 <-
957     _process_selector_ = proc1
958     proc1.running = TRUE
959     proc0.running = FALSE
960  -> State: 4.10 <-
961     semaphore = 2
962     apr = 2
963     proc1.state = critical1
964  -> Input: 4.11 <-
965     _process_selector_ = proc0
966     proc1.running = FALSE
967     proc0.running = TRUE
968  -> State: 4.11 <-
969     semaphore = 3
970     proc0.state = idle
971  -> Input: 4.12 <-
972     _process_selector_ = proc2
973     proc2.running = TRUE
974     proc0.running = FALSE
975  -> State: 4.12 <-
976     semaphore = 2
977     apr = 3
978     proc2.state = critical1
979  -> Input: 4.13 <-
980     _process_selector_ = proc3
981     proc3.running = TRUE
982     proc2.running = FALSE
983  -> State: 4.13 <-
```

```
984      semaphore = 1
985      apr = 4
986      proc3.state = critical1
987    -> Input: 4.14 <-
988      _process_selector_ = proc4
989      proc4.running = TRUE
990      proc3.running = FALSE
991    -> State: 4.14 <-
992      semaphore = 0
993      apr = 0
994      proc4.state = critical1
995    -> Input: 4.15 <-
996    -> State: 4.15 <-
997      proc4.state = critical2
998    -> Input: 4.16 <-
999    -> State: 4.16 <-
1000     proc4.state = exiting
1001   -> Input: 4.17 <-
1002   -> State: 4.17 <-
1003     semaphore = 1
1004     proc4.state = idle
1005   -> Input: 4.18 <-
1006     _process_selector_ = proc0
1007     proc4.running = FALSE
1008     proc0.running = TRUE
1009   -> State: 4.18 <-
1010     proc0.state = enqueue
1011   -> Input: 4.19 <-
1012   -> State: 4.19 <-
1013     semaphore = 0
1014     apr = 1
1015     proc0.state = critical1
1016   -> Input: 4.20 <-
1017   -> State: 4.20 <-
1018     proc0.state = critical2
1019   -> Input: 4.21 <-
1020     _process_selector_ = proc1
1021     proc1.running = TRUE
1022     proc0.running = FALSE
1023   -> State: 4.21 <-
1024     proc1.state = critical2
1025   -> Input: 4.22 <-
1026     _process_selector_ = proc2
1027     proc2.running = TRUE
1028     proc1.running = FALSE
1029   -> State: 4.22 <-
1030     proc2.state = critical2
1031   -> Input: 4.23 <-
1032     _process_selector_ = proc3
1033     proc3.running = TRUE
1034     proc2.running = FALSE
1035   -> State: 4.23 <-
1036     proc3.state = critical2
1037   -> Input: 4.24 <-
1038     _process_selector_ = proc4
1039     proc4.running = TRUE
1040     proc3.running = FALSE
1041   -- Loop starts here
1042   -> State: 4.24 <-
1043     proc4.state = enqueue
```

```
-> Input: 4.25 <-
    _process_selector_ = main
    running = TRUE
    proc4.running = FALSE
-- Loop starts here
-> State: 4.25 <-
-> Input: 4.26 <-
    _process_selector_ = proc0
    running = FALSE
    proc0.running = TRUE
-> State: 4.26 <-
    proc0.state = exiting
-> Input: 4.27 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
-> State: 4.27 <-
    proc1.state = exiting
-> Input: 4.28 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
-> State: 4.28 <-
    proc2.state = exiting
-> Input: 4.29 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
-> State: 4.29 <-
    proc3.state = exiting
-> Input: 4.30 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
-> State: 4.30 <-
-> Input: 4.31 <-
    _process_selector_ = proc3
    proc4.running = FALSE
    proc3.running = TRUE
-> State: 4.31 <-
    semaphore = 1
    proc3.state = idle
-> Input: 4.32 <-
    _process_selector_ = proc2
    proc3.running = FALSE
    proc2.running = TRUE
-> State: 4.32 <-
    semaphore = 2
    proc2.state = idle
-> Input: 4.33 <-
    _process_selector_ = proc1
    proc2.running = FALSE
    proc1.running = TRUE
-> State: 4.33 <-
    semaphore = 3
    proc1.state = idle
-> Input: 4.34 <-
-> State: 4.34 <-
    proc1.state = enqueue
-> Input: 4.35 <-
```

```
-> State: 4.35 <-
  semaphore = 2
  apr = 2
  proc1.state = critical1
-> Input: 4.36 <-
  _process_selector_ = proc0
  proc1.running = FALSE
  proc0.running = TRUE
-> State: 4.36 <-
  semaphore = 3
  proc0.state = idle
-> Input: 4.37 <-
  _process_selector_ = proc2
  proc2.running = TRUE
  proc0.running = FALSE
-> State: 4.37 <-
  proc2.state = enqueue
-> Input: 4.38 <-
  _process_selector_ = proc3
  proc3.running = TRUE
  proc2.running = FALSE
-> State: 4.38 <-
  proc3.state = enqueue
-> Input: 4.39 <-
  _process_selector_ = proc2
  proc3.running = FALSE
  proc2.running = TRUE
-> State: 4.39 <-
  semaphore = 2
  apr = 3
  proc2.state = critical1
-> Input: 4.40 <-
  _process_selector_ = proc3
  proc3.running = TRUE
  proc2.running = FALSE
-> State: 4.40 <-
  semaphore = 1
  apr = 4
  proc3.state = critical1
-> Input: 4.41 <-
  _process_selector_ = proc4
  proc4.running = TRUE
  proc3.running = FALSE
-> State: 4.41 <-
  semaphore = 0
  apr = 0
  proc4.state = critical1
-> Input: 4.42 <-
-> State: 4.42 <-
  proc4.state = critical2
-> Input: 4.43 <-
-> State: 4.43 <-
  proc4.state = exiting
-> Input: 4.44 <-
-> State: 4.44 <-
  semaphore = 1
  proc4.state = idle
-> Input: 4.45 <-
  _process_selector_ = proc0
  proc4.running = FALSE
```

```
1164      proc0.running = TRUE
1165    -> State: 4.45 <-
1166      proc0.state = enqueue
1167    -> Input: 4.46 <-
1168    -> State: 4.46 <-
1169      semaphore = 0
1170      apr = 1
1171      proc0.state = critical1
1172    -> Input: 4.47 <-
1173    -> State: 4.47 <-
1174      proc0.state = critical2
1175    -> Input: 4.48 <-
1176      _process_selector_ = proc1
1177      proc1.running = TRUE
1178      proc0.running = FALSE
1179    -> State: 4.48 <-
1180      proc1.state = critical2
1181    -> Input: 4.49 <-
1182      _process_selector_ = proc2
1183      proc2.running = TRUE
1184      proc1.running = FALSE
1185    -> State: 4.49 <-
1186      proc2.state = critical2
1187    -> Input: 4.50 <-
1188      _process_selector_ = proc4
1189      proc4.running = TRUE
1190      proc2.running = FALSE
1191    -> State: 4.50 <-
1192      proc4.state = enqueue
1193    -> Input: 4.51 <-
1194      _process_selector_ = proc3
1195      proc4.running = FALSE
1196      proc3.running = TRUE
1197    -> State: 4.51 <-
1198      proc3.state = critical2
1199  -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
      ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state =
      ↪ critical2))  is true
```

5.6. Приложение F

Результаты моделирования при инициализации семафора 5-й

Листинг 7: Инициализация семаформа - 5

```
1  *** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:56 2015)
2  *** Enabled addons are: compass
3  *** For more information on NuSMV see <http://nusmv.fbk.eu>
4  *** or email to <nusmv-users@list.fbk.eu>.
5  *** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>
6
7  *** Copyright (c) 2010-2014, Fondazione Bruno Kessler
8
9  *** This version of NuSMV is linked to the CUDD library version 2.4.1
10 *** Copyright (c) 1995-2004, Regents of the University of Colorado
11
12 *** This version of NuSMV is linked to the MiniSat SAT solver.
13 *** See http://minisat.se/MiniSat.html
14 *** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
15 *** Copyright (c) 2007-2010, Niklas Sorensson
```

```
16
17 WARNING *** The model contains PROCESSes or ISAs. ***
18 WARNING *** The HRC hierarchy will not be usable. ***
19 -- specification ((((( F proc0.state = critical2 &  F proc1.state = critical2) &
       ↪  F proc2.state = critical2) &  F proc3.state = critical2) &  F proc4.state
       ↪  = critical2)  is true
20 -- specification !( F ((((proc0.state = critical2 & proc1.state = idle) & proc2.
       ↪ state = idle) & proc3.state = idle) & proc4.state = idle))  is false
21 -- as demonstrated by the following execution sequence
22 Trace Description: LTL Counterexample
23 Trace Type: Counterexample
24   -> State: 1.1 <-
25     semaphore = 5
26     apr = 0
27     proc0.state = idle
28     proc1.state = idle
29     proc2.state = idle
30     proc3.state = idle
31     proc4.state = idle
32   -> Input: 1.2 <-
33     _process_selector_ = proc0
34     running = FALSE
35     proc4.running = FALSE
36     proc3.running = FALSE
37     proc2.running = FALSE
38     proc1.running = FALSE
39     proc0.running = TRUE
40   -> State: 1.2 <-
41     proc0.state = enqueue
42   -> Input: 1.3 <-
43     _process_selector_ = proc1
44     proc1.running = TRUE
45     proc0.running = FALSE
46   -> State: 1.3 <-
47     proc1.state = enqueue
48   -> Input: 1.4 <-
49     _process_selector_ = proc2
50     proc2.running = TRUE
51     proc1.running = FALSE
52   -> State: 1.4 <-
53     proc2.state = enqueue
54   -> Input: 1.5 <-
55     _process_selector_ = proc3
56     proc3.running = TRUE
57     proc2.running = FALSE
58   -> State: 1.5 <-
59     proc3.state = enqueue
60   -> Input: 1.6 <-
61     _process_selector_ = proc4
62     proc4.running = TRUE
63     proc3.running = FALSE
64   -> State: 1.6 <-
65     proc4.state = enqueue
66   -> Input: 1.7 <-
67     _process_selector_ = proc0
68     proc4.running = FALSE
69     proc0.running = TRUE
70   -> State: 1.7 <-
71     semaphore = 4
72     apr = 1
```

```
73      proc0.state = critical1
74    -> Input: 1.8 <-
75      _process_selector_ = proc1
76      proc1.running = TRUE
77      proc0.running = FALSE
78    -> State: 1.8 <-
79      semaphore = 3
80      apr = 2
81      proc1.state = critical1
82    -> Input: 1.9 <-
83      _process_selector_ = proc2
84      proc2.running = TRUE
85      proc1.running = FALSE
86    -> State: 1.9 <-
87      semaphore = 2
88      apr = 3
89      proc2.state = critical1
90    -> Input: 1.10 <-
91      _process_selector_ = proc3
92      proc3.running = TRUE
93      proc2.running = FALSE
94    -> State: 1.10 <-
95      semaphore = 1
96      apr = 4
97      proc3.state = critical1
98    -> Input: 1.11 <-
99      _process_selector_ = proc4
100     proc4.running = TRUE
101     proc3.running = FALSE
102   -> State: 1.11 <-
103     semaphore = 0
104     apr = 0
105     proc4.state = critical1
106   -> Input: 1.12 <-
107   -> State: 1.12 <-
108     proc4.state = critical2
109   -> Input: 1.13 <-
110   -> State: 1.13 <-
111     proc4.state = exiting
112   -> Input: 1.14 <-
113   -> State: 1.14 <-
114     semaphore = 1
115     proc4.state = idle
116   -> Input: 1.15 <-
117     _process_selector_ = proc2
118     proc4.running = FALSE
119     proc2.running = TRUE
120   -> State: 1.15 <-
121     proc2.state = critical2
122   -> Input: 1.16 <-
123     _process_selector_ = proc3
124     proc3.running = TRUE
125     proc2.running = FALSE
126   -> State: 1.16 <-
127     proc3.state = critical2
128   -> Input: 1.17 <-
129   -> State: 1.17 <-
130     proc3.state = exiting
131   -> Input: 1.18 <-
132     _process_selector_ = proc2
```

```
133     proc3.running = FALSE
134     proc2.running = TRUE
135   -> State: 1.18 <-
136     proc2.state = exiting
137   -> Input: 1.19 <-
138     _process_selector_ = proc3
139     proc3.running = TRUE
140     proc2.running = FALSE
141   -> State: 1.19 <-
142     semaphore = 2
143     proc3.state = idle
144   -> Input: 1.20 <-
145     _process_selector_ = proc2
146     proc3.running = FALSE
147     proc2.running = TRUE
148   -> State: 1.20 <-
149     semaphore = 3
150     proc2.state = idle
151   -> Input: 1.21 <-
152     _process_selector_ = proc0
153     proc2.running = FALSE
154     proc0.running = TRUE
155   -> State: 1.21 <-
156     proc0.state = critical2
157   -> Input: 1.22 <-
158     _process_selector_ = proc1
159     proc1.running = TRUE
160     proc0.running = FALSE
161   -> State: 1.22 <-
162     proc1.state = critical2
163   -> Input: 1.23 <-
164   -> State: 1.23 <-
165     proc1.state = exiting
166   -> Input: 1.24 <-
167   -> State: 1.24 <-
168     semaphore = 4
169     proc1.state = idle
170   -> Input: 1.25 <-
171     _process_selector_ = proc4
172     proc4.running = TRUE
173     proc1.running = FALSE
174   -> State: 1.25 <-
175     proc4.state = enqueue
176   -> Input: 1.26 <-
177     _process_selector_ = proc1
178     proc4.running = FALSE
179     proc1.running = TRUE
180   -- Loop starts here
181   -> State: 1.26 <-
182     proc1.state = enqueue
183   -> Input: 1.27 <-
184     _process_selector_ = proc0
185     proc1.running = FALSE
186     proc0.running = TRUE
187   -> State: 1.27 <-
188     proc0.state = exiting
189   -> Input: 1.28 <-
190     _process_selector_ = proc1
191     proc1.running = TRUE
192     proc0.running = FALSE
```

```
-> State: 1.28 <-
-> Input: 1.29 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
-> State: 1.29 <-
    proc2.state = enqueue
-> Input: 1.30 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
-> State: 1.30 <-
    proc3.state = enqueue
-> Input: 1.31 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc3.running = FALSE
-> State: 1.31 <-
-> Input: 1.32 <-
    _process_selector_ = proc0
    proc4.running = FALSE
    proc0.running = TRUE
-> State: 1.32 <-
    semaphore = 5
    proc0.state = idle
-> Input: 1.33 <-
-> State: 1.33 <-
    proc0.state = enqueue
-> Input: 1.34 <-
-> State: 1.34 <-
    semaphore = 4
    apr = 1
    proc0.state = critical1
-> Input: 1.35 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
-> State: 1.35 <-
    semaphore = 3
    apr = 2
    proc1.state = critical1
-> Input: 1.36 <-
    _process_selector_ = proc2
    proc2.running = TRUE
    proc1.running = FALSE
-> State: 1.36 <-
    semaphore = 2
    apr = 3
    proc2.state = critical1
-> Input: 1.37 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
-> State: 1.37 <-
    semaphore = 1
    apr = 4
    proc3.state = critical1
-> Input: 1.38 <-
    _process_selector_ = proc4
    proc4.running = TRUE
```

```
253     proc3.running = FALSE
254   -> State: 1.38 <-
255     semaphore = 0
256     apr = 0
257     proc4.state = critical1
258   -> Input: 1.39 <-
259   -> State: 1.39 <-
260     proc4.state = critical2
261   -> Input: 1.40 <-
262   -> State: 1.40 <-
263     proc4.state = exiting
264   -> Input: 1.41 <-
265   -> State: 1.41 <-
266     semaphore = 1
267     proc4.state = idle
268   -> Input: 1.42 <-
269     _process_selector_ = proc1
270     proc4.running = FALSE
271     proc1.running = TRUE
272   -> State: 1.42 <-
273     proc1.state = critical2
274   -> Input: 1.43 <-
275     _process_selector_ = proc3
276     proc3.running = TRUE
277     proc1.running = FALSE
278   -> State: 1.43 <-
279     proc3.state = critical2
280   -> Input: 1.44 <-
281   -> State: 1.44 <-
282     proc3.state = exiting
283   -> Input: 1.45 <-
284     _process_selector_ = proc1
285     proc3.running = FALSE
286     proc1.running = TRUE
287   -> State: 1.45 <-
288     proc1.state = exiting
289   -> Input: 1.46 <-
290     _process_selector_ = proc3
291     proc3.running = TRUE
292     proc1.running = FALSE
293   -> State: 1.46 <-
294     semaphore = 2
295     proc3.state = idle
296   -> Input: 1.47 <-
297     _process_selector_ = proc1
298     proc3.running = FALSE
299     proc1.running = TRUE
300   -> State: 1.47 <-
301     semaphore = 3
302     proc1.state = idle
303   -> Input: 1.48 <-
304     _process_selector_ = proc0
305     proc1.running = FALSE
306     proc0.running = TRUE
307   -> State: 1.48 <-
308     proc0.state = critical2
309   -> Input: 1.49 <-
310     _process_selector_ = proc1
311     proc1.running = TRUE
312     proc0.running = FALSE
```

```
313   -> State: 1.49 <-
314     proc1.state = enqueue
315   -> Input: 1.50 <-
316     _process_selector_ = proc2
317     proc2.running = TRUE
318     proc1.running = FALSE
319   -> State: 1.50 <-
320     proc2.state = critical2
321   -> Input: 1.51 <-
322     _process_selector_ = proc4
323     proc4.running = TRUE
324     proc2.running = FALSE
325   -> State: 1.51 <-
326     proc4.state = enqueue
327   -> Input: 1.52 <-
328     _process_selector_ = proc2
329     proc4.running = FALSE
330     proc2.running = TRUE
331   -> State: 1.52 <-
332     proc2.state = exiting
333   -> Input: 1.53 <-
334   -> State: 1.53 <-
335     semaphore = 4
336     proc2.state = idle
337 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
      ↪ proc2.state = idle) & proc3.state = idle) & proc4.state = idle))  is false
338 -- as demonstrated by the following execution sequence
339 Trace Description: LTL Counterexample
340 Trace Type: Counterexample
341   -> State: 2.1 <-
342     semaphore = 5
343     apr = 0
344     proc0.state = idle
345     proc1.state = idle
346     proc2.state = idle
347     proc3.state = idle
348     proc4.state = idle
349   -> Input: 2.2 <-
350     _process_selector_ = proc0
351     running = FALSE
352     proc4.running = FALSE
353     proc3.running = FALSE
354     proc2.running = FALSE
355     proc1.running = FALSE
356     proc0.running = TRUE
357   -> State: 2.2 <-
358     proc0.state = enqueue
359   -> Input: 2.3 <-
360     _process_selector_ = proc1
361     proc1.running = TRUE
362     proc0.running = FALSE
363   -> State: 2.3 <-
364     proc1.state = enqueue
365   -> Input: 2.4 <-
366     _process_selector_ = proc2
367     proc2.running = TRUE
368     proc1.running = FALSE
369   -> State: 2.4 <-
370     proc2.state = enqueue
371   -> Input: 2.5 <-
```

```
372      _process_selector_ = proc3
373      proc3.running = TRUE
374      proc2.running = FALSE
375    -> State: 2.5 <-
376      proc3.state = enqueue
377    -> Input: 2.6 <-
378      _process_selector_ = proc4
379      proc4.running = TRUE
380      proc3.running = FALSE
381    -> State: 2.6 <-
382      proc4.state = enqueue
383    -> Input: 2.7 <-
384      _process_selector_ = proc0
385      proc4.running = FALSE
386      proc0.running = TRUE
387    -> State: 2.7 <-
388      semaphore = 4
389      apr = 1
390      proc0.state = critical1
391    -> Input: 2.8 <-
392      _process_selector_ = proc1
393      proc1.running = TRUE
394      proc0.running = FALSE
395    -> State: 2.8 <-
396      semaphore = 3
397      apr = 2
398      proc1.state = critical1
399    -> Input: 2.9 <-
400      _process_selector_ = proc2
401      proc2.running = TRUE
402      proc1.running = FALSE
403    -> State: 2.9 <-
404      semaphore = 2
405      apr = 3
406      proc2.state = critical1
407    -> Input: 2.10 <-
408      _process_selector_ = proc3
409      proc3.running = TRUE
410      proc2.running = FALSE
411    -> State: 2.10 <-
412      semaphore = 1
413      apr = 4
414      proc3.state = critical1
415    -> Input: 2.11 <-
416      _process_selector_ = proc4
417      proc4.running = TRUE
418      proc3.running = FALSE
419    -> State: 2.11 <-
420      semaphore = 0
421      apr = 0
422      proc4.state = critical1
423    -> Input: 2.12 <-
424    -> State: 2.12 <-
425      proc4.state = critical2
426    -> Input: 2.13 <-
427    -> State: 2.13 <-
428      proc4.state = exiting
429    -> Input: 2.14 <-
430    -> State: 2.14 <-
431      semaphore = 1
```

```
    proc4.state = idle
  -> Input: 2.15 <-
    _process_selector_ = proc2
    proc4.running = FALSE
    proc2.running = TRUE
  -> State: 2.15 <-
    proc2.state = critical2
  -> Input: 2.16 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 2.16 <-
    proc3.state = critical2
  -> Input: 2.17 <-
  -> State: 2.17 <-
    proc3.state = exiting
  -> Input: 2.18 <-
    _process_selector_ = proc2
    proc3.running = FALSE
    proc2.running = TRUE
  -> State: 2.18 <-
    proc2.state = exiting
  -> Input: 2.19 <-
    _process_selector_ = proc3
    proc3.running = TRUE
    proc2.running = FALSE
  -> State: 2.19 <-
    semaphore = 2
    proc3.state = idle
  -> Input: 2.20 <-
    _process_selector_ = proc2
    proc3.running = FALSE
    proc2.running = TRUE
  -> State: 2.20 <-
    semaphore = 3
    proc2.state = idle
  -> Input: 2.21 <-
    _process_selector_ = proc0
    proc2.running = FALSE
    proc0.running = TRUE
  -> State: 2.21 <-
    proc0.state = critical2
  -> Input: 2.22 <-
    _process_selector_ = proc1
    proc1.running = TRUE
    proc0.running = FALSE
  -> State: 2.22 <-
    proc1.state = critical2
  -> Input: 2.23 <-
    _process_selector_ = proc4
    proc4.running = TRUE
    proc1.running = FALSE
  -> State: 2.23 <-
    proc4.state = enqueue
  -> Input: 2.24 <-
    _process_selector_ = proc2
    proc4.running = FALSE
    proc2.running = TRUE
  -- Loop starts here
  -> State: 2.24 <-
```

```
492     proc2.state = enqueue
493   -> Input: 2.25 <-
494     _process_selector_ = proc0
495     proc2.running = FALSE
496     proc0.running = TRUE
497   -> State: 2.25 <-
498     proc0.state = exiting
499   -> Input: 2.26 <-
500     _process_selector_ = proc1
501     proc1.running = TRUE
502     proc0.running = FALSE
503   -> State: 2.26 <-
504     proc1.state = exiting
505   -> Input: 2.27 <-
506     _process_selector_ = proc2
507     proc2.running = TRUE
508     proc1.running = FALSE
509   -> State: 2.27 <-
510   -> Input: 2.28 <-
511     _process_selector_ = proc3
512     proc3.running = TRUE
513     proc2.running = FALSE
514   -> State: 2.28 <-
515     proc3.state = enqueue
516   -> Input: 2.29 <-
517     _process_selector_ = proc4
518     proc4.running = TRUE
519     proc3.running = FALSE
520   -> State: 2.29 <-
521   -> Input: 2.30 <-
522     _process_selector_ = proc1
523     proc4.running = FALSE
524     proc1.running = TRUE
525   -> State: 2.30 <-
526     semaphore = 4
527     proc1.state = idle
528   -> Input: 2.31 <-
529     _process_selector_ = proc0
530     proc1.running = FALSE
531     proc0.running = TRUE
532   -> State: 2.31 <-
533     semaphore = 5
534     proc0.state = idle
535   -> Input: 2.32 <-
536   -> State: 2.32 <-
537     proc0.state = enqueue
538   -> Input: 2.33 <-
539     _process_selector_ = proc1
540     proc1.running = TRUE
541     proc0.running = FALSE
542   -> State: 2.33 <-
543     proc1.state = enqueue
544   -> Input: 2.34 <-
545     _process_selector_ = proc0
546     proc1.running = FALSE
547     proc0.running = TRUE
548   -> State: 2.34 <-
549     semaphore = 4
550     apr = 1
551     proc0.state = critical1
```

```
552  -> Input: 2.35 <-
553     _process_selector_ = proc1
554     proc1.running = TRUE
555     proc0.running = FALSE
556  -> State: 2.35 <-
557     semaphore = 3
558     apr = 2
559     proc1.state = critical1
560  -> Input: 2.36 <-
561     _process_selector_ = proc2
562     proc2.running = TRUE
563     proc1.running = FALSE
564  -> State: 2.36 <-
565     semaphore = 2
566     apr = 3
567     proc2.state = critical1
568  -> Input: 2.37 <-
569     _process_selector_ = proc3
570     proc3.running = TRUE
571     proc2.running = FALSE
572  -> State: 2.37 <-
573     semaphore = 1
574     apr = 4
575     proc3.state = critical1
576  -> Input: 2.38 <-
577     _process_selector_ = proc4
578     proc4.running = TRUE
579     proc3.running = FALSE
580  -> State: 2.38 <-
581     semaphore = 0
582     apr = 0
583     proc4.state = critical1
584  -> Input: 2.39 <-
585  -> State: 2.39 <-
586     proc4.state = critical2
587  -> Input: 2.40 <-
588  -> State: 2.40 <-
589     proc4.state = exiting
590  -> Input: 2.41 <-
591  -> State: 2.41 <-
592     semaphore = 1
593     proc4.state = idle
594  -> Input: 2.42 <-
595     _process_selector_ = proc2
596     proc4.running = FALSE
597     proc2.running = TRUE
598  -> State: 2.42 <-
599     proc2.state = critical2
600  -> Input: 2.43 <-
601     _process_selector_ = proc3
602     proc3.running = TRUE
603     proc2.running = FALSE
604  -> State: 2.43 <-
605     proc3.state = critical2
606  -> Input: 2.44 <-
607  -> State: 2.44 <-
608     proc3.state = exiting
609  -> Input: 2.45 <-
610     _process_selector_ = proc2
611     proc3.running = FALSE
```

```
612     proc2.running = TRUE
613   -> State: 2.45 <-
614     proc2.state = exiting
615   -> Input: 2.46 <-
616     _process_selector_ = proc3
617     proc3.running = TRUE
618     proc2.running = FALSE
619   -> State: 2.46 <-
620     semaphore = 2
621     proc3.state = idle
622   -> Input: 2.47 <-
623     _process_selector_ = proc2
624     proc3.running = FALSE
625     proc2.running = TRUE
626   -> State: 2.47 <-
627     semaphore = 3
628     proc2.state = idle
629   -> Input: 2.48 <-
630     _process_selector_ = proc0
631     proc2.running = FALSE
632     proc0.running = TRUE
633   -> State: 2.48 <-
634     proc0.state = critical2
635   -> Input: 2.49 <-
636     _process_selector_ = proc2
637     proc2.running = TRUE
638     proc0.running = FALSE
639   -> State: 2.49 <-
640     proc2.state = enqueue
641   -> Input: 2.50 <-
642     _process_selector_ = proc1
643     proc2.running = FALSE
644     proc1.running = TRUE
645   -> State: 2.50 <-
646     proc1.state = critical2
647   -> Input: 2.51 <-
648     _process_selector_ = proc4
649     proc4.running = TRUE
650     proc1.running = FALSE
651   -> State: 2.51 <-
652     proc4.state = enqueue
653 -- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
      ↪ proc2.state = critical2) & proc3.state = idle) & proc4.state = idle))  is
      ↪ false
654 -- as demonstrated by the following execution sequence
655 Trace Description: LTL Counterexample
656 Trace Type: Counterexample
657   -> State: 3.1 <-
658     semaphore = 5
659     apr = 0
660     proc0.state = idle
661     proc1.state = idle
662     proc2.state = idle
663     proc3.state = idle
664     proc4.state = idle
665   -> Input: 3.2 <-
666     _process_selector_ = proc0
667     running = FALSE
668     proc4.running = FALSE
669     proc3.running = FALSE
```

```
670     proc2.running = FALSE
671     proc1.running = FALSE
672     proc0.running = TRUE
673   -> State: 3.2 <-
674     proc0.state = enqueue
675   -> Input: 3.3 <-
676     _process_selector_ = proc1
677     proc1.running = TRUE
678     proc0.running = FALSE
679   -> State: 3.3 <-
680     proc1.state = enqueue
681   -> Input: 3.4 <-
682     _process_selector_ = proc2
683     proc2.running = TRUE
684     proc1.running = FALSE
685   -> State: 3.4 <-
686     proc2.state = enqueue
687   -> Input: 3.5 <-
688     _process_selector_ = proc3
689     proc3.running = TRUE
690     proc2.running = FALSE
691   -> State: 3.5 <-
692     proc3.state = enqueue
693   -> Input: 3.6 <-
694     _process_selector_ = proc4
695     proc4.running = TRUE
696     proc3.running = FALSE
697   -> State: 3.6 <-
698     proc4.state = enqueue
699   -> Input: 3.7 <-
700     _process_selector_ = proc0
701     proc4.running = FALSE
702     proc0.running = TRUE
703   -> State: 3.7 <-
704     semaphore = 4
705     apr = 1
706     proc0.state = critical1
707   -> Input: 3.8 <-
708     _process_selector_ = proc1
709     proc1.running = TRUE
710     proc0.running = FALSE
711   -> State: 3.8 <-
712     semaphore = 3
713     apr = 2
714     proc1.state = critical1
715   -> Input: 3.9 <-
716     _process_selector_ = proc2
717     proc2.running = TRUE
718     proc1.running = FALSE
719   -> State: 3.9 <-
720     semaphore = 2
721     apr = 3
722     proc2.state = critical1
723   -> Input: 3.10 <-
724     _process_selector_ = proc3
725     proc3.running = TRUE
726     proc2.running = FALSE
727   -> State: 3.10 <-
728     semaphore = 1
729     apr = 4
```

```
730     proc3.state = critical1
731   -> Input: 3.11 <-
732     _process_selector_ = proc4
733     proc4.running = TRUE
734     proc3.running = FALSE
735   -> State: 3.11 <-
736     semaphore = 0
737     apr = 0
738     proc4.state = critical1
739   -> Input: 3.12 <-
740   -> State: 3.12 <-
741     proc4.state = critical2
742   -> Input: 3.13 <-
743   -> State: 3.13 <-
744     proc4.state = exiting
745   -> Input: 3.14 <-
746   -> State: 3.14 <-
747     semaphore = 1
748     proc4.state = idle
749   -> Input: 3.15 <-
750     _process_selector_ = proc0
751     proc4.running = FALSE
752     proc0.running = TRUE
753   -> State: 3.15 <-
754     proc0.state = critical2
755   -> Input: 3.16 <-
756     _process_selector_ = proc1
757     proc1.running = TRUE
758     proc0.running = FALSE
759   -> State: 3.16 <-
760     proc1.state = critical2
761   -> Input: 3.17 <-
762     _process_selector_ = proc2
763     proc2.running = TRUE
764     proc1.running = FALSE
765   -> State: 3.17 <-
766     proc2.state = critical2
767   -> Input: 3.18 <-
768     _process_selector_ = proc3
769     proc3.running = TRUE
770     proc2.running = FALSE
771   -> State: 3.18 <-
772     proc3.state = critical2
773   -> Input: 3.19 <-
774   -> State: 3.19 <-
775     proc3.state = exiting
776   -> Input: 3.20 <-
777   -> State: 3.20 <-
778     semaphore = 2
779     proc3.state = idle
780   -> Input: 3.21 <-
781     _process_selector_ = proc4
782     proc4.running = TRUE
783     proc3.running = FALSE
784   -> State: 3.21 <-
785     proc4.state = enqueue
786   -> Input: 3.22 <-
787     _process_selector_ = proc3
788     proc4.running = FALSE
789     proc3.running = TRUE
```

```
790   -- Loop starts here
791   -> State: 3.22 <-
792     proc3.state = enqueue
793   -> Input: 3.23 <-
794     _process_selector_ = proc0
795     proc3.running = FALSE
796     proc0.running = TRUE
797   -> State: 3.23 <-
798     proc0.state = exiting
799   -> Input: 3.24 <-
800     _process_selector_ = proc1
801     proc1.running = TRUE
802     proc0.running = FALSE
803   -> State: 3.24 <-
804     proc1.state = exiting
805   -> Input: 3.25 <-
806     _process_selector_ = proc2
807     proc2.running = TRUE
808     proc1.running = FALSE
809   -> State: 3.25 <-
810     proc2.state = exiting
811   -> Input: 3.26 <-
812     _process_selector_ = proc3
813     proc3.running = TRUE
814     proc2.running = FALSE
815   -> State: 3.26 <-
816   -> Input: 3.27 <-
817     _process_selector_ = proc4
818     proc4.running = TRUE
819     proc3.running = FALSE
820   -> State: 3.27 <-
821   -> Input: 3.28 <-
822     _process_selector_ = proc0
823     proc4.running = FALSE
824     proc0.running = TRUE
825   -> State: 3.28 <-
826     semaphore = 3
827     proc0.state = idle
828   -> Input: 3.29 <-
829   -> State: 3.29 <-
830     proc0.state = enqueue
831   -> Input: 3.30 <-
832   -> State: 3.30 <-
833     semaphore = 2
834     apr = 1
835     proc0.state = critical1
836   -> Input: 3.31 <-
837     _process_selector_ = proc1
838     proc1.running = TRUE
839     proc0.running = FALSE
840   -> State: 3.31 <-
841     semaphore = 3
842     proc1.state = idle
843   -> Input: 3.32 <-
844   -> State: 3.32 <-
845     proc1.state = enqueue
846   -> Input: 3.33 <-
847   -> State: 3.33 <-
848     semaphore = 2
849     apr = 2
```

```
850     proc1.state = critical1
851   -> Input: 3.34 <-
852     _process_selector_ = proc2
853     proc2.running = TRUE
854     proc1.running = FALSE
855   -> State: 3.34 <-
856     semaphore = 3
857     proc2.state = idle
858   -> Input: 3.35 <-
859   -> State: 3.35 <-
860     proc2.state = enqueue
861   -> Input: 3.36 <-
862   -> State: 3.36 <-
863     semaphore = 2
864     apr = 3
865     proc2.state = critical1
866   -> Input: 3.37 <-
867     _process_selector_ = proc3
868     proc3.running = TRUE
869     proc2.running = FALSE
870   -> State: 3.37 <-
871     semaphore = 1
872     apr = 4
873     proc3.state = critical1
874   -> Input: 3.38 <-
875   -> State: 3.38 <-
876     proc3.state = critical2
877   -> Input: 3.39 <-
878   -> State: 3.39 <-
879     proc3.state = exiting
880   -> Input: 3.40 <-
881   -> State: 3.40 <-
882     semaphore = 2
883     proc3.state = idle
884   -> Input: 3.41 <-
885     _process_selector_ = proc4
886     proc4.running = TRUE
887     proc3.running = FALSE
888   -> State: 3.41 <-
889     semaphore = 1
890     apr = 0
891     proc4.state = critical1
892   -> Input: 3.42 <-
893     _process_selector_ = proc0
894     proc4.running = FALSE
895     proc0.running = TRUE
896   -> State: 3.42 <-
897     proc0.state = critical2
898   -> Input: 3.43 <-
899     _process_selector_ = proc1
900     proc1.running = TRUE
901     proc0.running = FALSE
902   -> State: 3.43 <-
903     proc1.state = critical2
904   -> Input: 3.44 <-
905     _process_selector_ = proc2
906     proc2.running = TRUE
907     proc1.running = FALSE
908   -> State: 3.44 <-
909     proc2.state = critical2
```

```
  -> Input: 3.45 <-
     _process_selector_ = proc3
     proc3.running = TRUE
     proc2.running = FALSE
  -> State: 3.45 <-
     proc3.state = enqueue
  -> Input: 3.46 <-
     _process_selector_ = proc4
     proc4.running = TRUE
     proc3.running = FALSE
  -> State: 3.46 <-
     proc4.state = critical2
  -> Input: 3.47 <-
  -> State: 3.47 <-
     proc4.state = exiting
  -> Input: 3.48 <-
  -> State: 3.48 <-
     semaphore = 2
     proc4.state = idle
  -> Input: 3.49 <-
  -> State: 3.49 <-
     proc4.state = enqueue
-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
   ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state = idle))
   ↪    is false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
Trace Type: Counterexample
  -> State: 4.1 <-
     semaphore = 5
     apr = 0
     proc0.state = idle
     proc1.state = idle
     proc2.state = idle
     proc3.state = idle
     proc4.state = idle
  -> Input: 4.2 <-
     _process_selector_ = proc0
     running = FALSE
     proc4.running = FALSE
     proc3.running = FALSE
     proc2.running = FALSE
     proc1.running = FALSE
     proc0.running = TRUE
  -> State: 4.2 <-
     proc0.state = enqueue
  -> Input: 4.3 <-
     _process_selector_ = proc1
     proc1.running = TRUE
     proc0.running = FALSE
  -> State: 4.3 <-
     proc1.state = enqueue
  -> Input: 4.4 <-
     _process_selector_ = proc2
     proc2.running = TRUE
     proc1.running = FALSE
  -> State: 4.4 <-
     proc2.state = enqueue
  -> Input: 4.5 <-
     _process_selector_ = proc3
```

```
968     proc3.running = TRUE
969     proc2.running = FALSE
970   -> State: 4.5 <-
971     proc3.state = enqueue
972   -> Input: 4.6 <-
973     _process_selector_ = proc4
974     proc4.running = TRUE
975     proc3.running = FALSE
976   -> State: 4.6 <-
977     proc4.state = enqueue
978   -> Input: 4.7 <-
979     _process_selector_ = proc0
980     proc4.running = FALSE
981     proc0.running = TRUE
982   -> State: 4.7 <-
983     semaphore = 4
984     apr = 1
985     proc0.state = critical1
986   -> Input: 4.8 <-
987     _process_selector_ = proc1
988     proc1.running = TRUE
989     proc0.running = FALSE
990   -> State: 4.8 <-
991     semaphore = 3
992     apr = 2
993     proc1.state = critical1
994   -> Input: 4.9 <-
995     _process_selector_ = proc2
996     proc2.running = TRUE
997     proc1.running = FALSE
998   -> State: 4.9 <-
999     semaphore = 2
1000    apr = 3
1001    proc2.state = critical1
1002  -> Input: 4.10 <-
1003    _process_selector_ = proc3
1004    proc3.running = TRUE
1005    proc2.running = FALSE
1006  -> State: 4.10 <-
1007    semaphore = 1
1008    apr = 4
1009    proc3.state = critical1
1010  -> Input: 4.11 <-
1011    _process_selector_ = proc4
1012    proc4.running = TRUE
1013    proc3.running = FALSE
1014  -> State: 4.11 <-
1015    semaphore = 0
1016    apr = 0
1017    proc4.state = critical1
1018  -> Input: 4.12 <-
1019  -> State: 4.12 <-
1020    proc4.state = critical2
1021  -> Input: 4.13 <-
1022  -> State: 4.13 <-
1023    proc4.state = exiting
1024  -> Input: 4.14 <-
1025  -> State: 4.14 <-
1026    semaphore = 1
1027    proc4.state = idle
```

```
1028  -> Input: 4.15 <-
1029     _process_selector_ = proc0
1030     proc4.running = FALSE
1031     proc0.running = TRUE
1032  -> State: 4.15 <-
1033     proc0.state = critical2
1034  -> Input: 4.16 <-
1035     _process_selector_ = proc1
1036     proc1.running = TRUE
1037     proc0.running = FALSE
1038  -> State: 4.16 <-
1039     proc1.state = critical2
1040  -> Input: 4.17 <-
1041     _process_selector_ = proc2
1042     proc2.running = TRUE
1043     proc1.running = FALSE
1044  -> State: 4.17 <-
1045     proc2.state = critical2
1046  -> Input: 4.18 <-
1047     _process_selector_ = proc3
1048     proc3.running = TRUE
1049     proc2.running = FALSE
1050  -> State: 4.18 <-
1051     proc3.state = critical2
1052  -> Input: 4.19 <-
1053     _process_selector_ = proc4
1054     proc4.running = TRUE
1055     proc3.running = FALSE
1056  -- Loop starts here
1057  -> State: 4.19 <-
1058     proc4.state = enqueue
1059  -> Input: 4.20 <-
1060     _process_selector_ = main
1061     running = TRUE
1062     proc4.running = FALSE
1063  -- Loop starts here
1064  -> State: 4.20 <-
1065  -> Input: 4.21 <-
1066     _process_selector_ = proc0
1067     running = FALSE
1068     proc0.running = TRUE
1069  -> State: 4.21 <-
1070     proc0.state = exiting
1071  -> Input: 4.22 <-
1072     _process_selector_ = proc1
1073     proc1.running = TRUE
1074     proc0.running = FALSE
1075  -> State: 4.22 <-
1076     proc1.state = exiting
1077  -> Input: 4.23 <-
1078     _process_selector_ = proc2
1079     proc2.running = TRUE
1080     proc1.running = FALSE
1081  -> State: 4.23 <-
1082     proc2.state = exiting
1083  -> Input: 4.24 <-
1084     _process_selector_ = proc3
1085     proc3.running = TRUE
1086     proc2.running = FALSE
1087  -> State: 4.24 <-
```

```
1088      proc3.state = exiting
1089    -> Input: 4.25 <-
1090      _process_selector_ = proc4
1091      proc4.running = TRUE
1092      proc3.running = FALSE
1093    -> State: 4.25 <-
1094    -> Input: 4.26 <-
1095      _process_selector_ = proc3
1096      proc4.running = FALSE
1097      proc3.running = TRUE
1098    -> State: 4.26 <-
1099      semaphore = 2
1100      proc3.state = idle
1101    -> Input: 4.27 <-
1102      _process_selector_ = proc0
1103      proc3.running = FALSE
1104      proc0.running = TRUE
1105    -> State: 4.27 <-
1106      semaphore = 3
1107      proc0.state = idle
1108    -> Input: 4.28 <-
1109    -> State: 4.28 <-
1110      proc0.state = enqueue
1111    -> Input: 4.29 <-
1112    -> State: 4.29 <-
1113      semaphore = 2
1114      apr = 1
1115      proc0.state = critical1
1116    -> Input: 4.30 <-
1117      _process_selector_ = proc1
1118      proc1.running = TRUE
1119      proc0.running = FALSE
1120    -> State: 4.30 <-
1121      semaphore = 3
1122      proc1.state = idle
1123    -> Input: 4.31 <-
1124    -> State: 4.31 <-
1125      proc1.state = enqueue
1126    -> Input: 4.32 <-
1127    -> State: 4.32 <-
1128      semaphore = 2
1129      apr = 2
1130      proc1.state = critical1
1131    -> Input: 4.33 <-
1132      _process_selector_ = proc2
1133      proc2.running = TRUE
1134      proc1.running = FALSE
1135    -> State: 4.33 <-
1136      semaphore = 3
1137      proc2.state = idle
1138    -> Input: 4.34 <-
1139    -> State: 4.34 <-
1140      proc2.state = enqueue
1141    -> Input: 4.35 <-
1142      _process_selector_ = proc3
1143      proc3.running = TRUE
1144      proc2.running = FALSE
1145    -> State: 4.35 <-
1146      proc3.state = enqueue
1147    -> Input: 4.36 <-
```

```
1148      _process_selector_ = proc2
1149      proc3.running = FALSE
1150      proc2.running = TRUE
1151    -> State: 4.36 <-
1152      semaphore = 2
1153      apr = 3
1154      proc2.state = critical1
1155    -> Input: 4.37 <-
1156      _process_selector_ = proc3
1157      proc3.running = TRUE
1158      proc2.running = FALSE
1159    -> State: 4.37 <-
1160      semaphore = 1
1161      apr = 4
1162      proc3.state = critical1
1163    -> Input: 4.38 <-
1164      _process_selector_ = proc4
1165      proc4.running = TRUE
1166      proc3.running = FALSE
1167    -> State: 4.38 <-
1168      semaphore = 0
1169      apr = 0
1170      proc4.state = critical1
1171    -> Input: 4.39 <-
1172    -> State: 4.39 <-
1173      proc4.state = critical2
1174    -> Input: 4.40 <-
1175    -> State: 4.40 <-
1176      proc4.state = exiting
1177    -> Input: 4.41 <-
1178    -> State: 4.41 <-
1179      semaphore = 1
1180      proc4.state = idle
1181    -> Input: 4.42 <-
1182      _process_selector_ = proc0
1183      proc4.running = FALSE
1184      proc0.running = TRUE
1185    -> State: 4.42 <-
1186      proc0.state = critical2
1187    -> Input: 4.43 <-
1188      _process_selector_ = proc1
1189      proc1.running = TRUE
1190      proc0.running = FALSE
1191    -> State: 4.43 <-
1192      proc1.state = critical2
1193    -> Input: 4.44 <-
1194      _process_selector_ = proc2
1195      proc2.running = TRUE
1196      proc1.running = FALSE
1197    -> State: 4.44 <-
1198      proc2.state = critical2
1199    -> Input: 4.45 <-
1200      _process_selector_ = proc4
1201      proc4.running = TRUE
1202      proc2.running = FALSE
1203    -> State: 4.45 <-
1204      proc4.state = enqueue
1205    -> Input: 4.46 <-
1206      _process_selector_ = proc3
1207      proc4.running = FALSE
```

```
1208|     proc3.running = TRUE
1209|   -> State: 4.46 <-
1210|     proc3.state = critical2
1211|-- specification !( F ((((proc0.state = critical2 & proc1.state = critical2) &
    |    ↪ proc2.state = critical2) & proc3.state = critical2) & proc4.state =
    |    ↪ critical2))  is false
1212|-- as demonstrated by the following execution sequence
1213|Trace Description: LTL Counterexample
1214|Trace Type: Counterexample
1215|   -> State: 5.1 <-
1216|     semaphore = 5
1217|     apr = 0
1218|     proc0.state = idle
1219|     proc1.state = idle
1220|     proc2.state = idle
1221|     proc3.state = idle
1222|     proc4.state = idle
1223|   -> Input: 5.2 <-
1224|     _process_selector_ = proc0
1225|     running = FALSE
1226|     proc4.running = FALSE
1227|     proc3.running = FALSE
1228|     proc2.running = FALSE
1229|     proc1.running = FALSE
1230|     proc0.running = TRUE
1231|   -> State: 5.2 <-
1232|     proc0.state = enqueue
1233|   -> Input: 5.3 <-
1234|     _process_selector_ = proc1
1235|     proc1.running = TRUE
1236|     proc0.running = FALSE
1237|   -> State: 5.3 <-
1238|     proc1.state = enqueue
1239|   -> Input: 5.4 <-
1240|     _process_selector_ = proc2
1241|     proc2.running = TRUE
1242|     proc1.running = FALSE
1243|   -> State: 5.4 <-
1244|     proc2.state = enqueue
1245|   -> Input: 5.5 <-
1246|     _process_selector_ = proc3
1247|     proc3.running = TRUE
1248|     proc2.running = FALSE
1249|   -> State: 5.5 <-
1250|     proc3.state = enqueue
1251|   -> Input: 5.6 <-
1252|     _process_selector_ = proc4
1253|     proc4.running = TRUE
1254|     proc3.running = FALSE
1255|   -> State: 5.6 <-
1256|     proc4.state = enqueue
1257|   -> Input: 5.7 <-
1258|     _process_selector_ = proc0
1259|     proc4.running = FALSE
1260|     proc0.running = TRUE
1261|   -> State: 5.7 <-
1262|     semaphore = 4
1263|     apr = 1
1264|     proc0.state = critical1
1265|   -> Input: 5.8 <-
```

```
1266      _process_selector_ = proc1
1267      proc1.running = TRUE
1268      proc0.running = FALSE
1269    -> State: 5.8 <-
1270      semaphore = 3
1271      apr = 2
1272      proc1.state = critical1
1273    -> Input: 5.9 <-
1274      _process_selector_ = proc2
1275      proc2.running = TRUE
1276      proc1.running = FALSE
1277    -> State: 5.9 <-
1278      semaphore = 2
1279      apr = 3
1280      proc2.state = critical1
1281    -> Input: 5.10 <-
1282      _process_selector_ = proc3
1283      proc3.running = TRUE
1284      proc2.running = FALSE
1285    -> State: 5.10 <-
1286      semaphore = 1
1287      apr = 4
1288      proc3.state = critical1
1289    -> Input: 5.11 <-
1290      _process_selector_ = proc0
1291      proc3.running = FALSE
1292      proc0.running = TRUE
1293    -> State: 5.11 <-
1294      proc0.state = critical2
1295    -> Input: 5.12 <-
1296      _process_selector_ = proc1
1297      proc1.running = TRUE
1298      proc0.running = FALSE
1299    -> State: 5.12 <-
1300      proc1.state = critical2
1301    -> Input: 5.13 <-
1302      _process_selector_ = proc2
1303      proc2.running = TRUE
1304      proc1.running = FALSE
1305    -> State: 5.13 <-
1306      proc2.state = critical2
1307    -> Input: 5.14 <-
1308      _process_selector_ = proc3
1309      proc3.running = TRUE
1310      proc2.running = FALSE
1311    -> State: 5.14 <-
1312      proc3.state = critical2
1313    -> Input: 5.15 <-
1314      _process_selector_ = proc4
1315      proc4.running = TRUE
1316      proc3.running = FALSE
1317    -> State: 5.15 <-
1318      semaphore = 0
1319      apr = 0
1320      proc4.state = critical1
1321    -> Input: 5.16 <-
1322    -> State: 5.16 <-
1323      proc4.state = critical2
1324    -> Input: 5.17 <-
1325    -> State: 5.17 <-
```

```
1326      proc4.state = exiting
1327    -> Input: 5.18 <-
1328    -- Loop starts here
1329    -> State: 5.18 <-
1330      semaphore = 1
1331      proc4.state = idle
1332    -> Input: 5.19 <-
1333      _process_selector_ = proc0
1334      proc4.running = FALSE
1335      proc0.running = TRUE
1336    -> State: 5.19 <-
1337      proc0.state = exiting
1338    -> Input: 5.20 <-
1339      _process_selector_ = proc1
1340      proc1.running = TRUE
1341      proc0.running = FALSE
1342    -> State: 5.20 <-
1343      proc1.state = exiting
1344    -> Input: 5.21 <-
1345      _process_selector_ = proc2
1346      proc2.running = TRUE
1347      proc1.running = FALSE
1348    -> State: 5.21 <-
1349      proc2.state = exiting
1350    -> Input: 5.22 <-
1351      _process_selector_ = proc3
1352      proc3.running = TRUE
1353      proc2.running = FALSE
1354    -> State: 5.22 <-
1355      proc3.state = exiting
1356    -> Input: 5.23 <-
1357      _process_selector_ = proc4
1358      proc4.running = TRUE
1359      proc3.running = FALSE
1360    -> State: 5.23 <-
1361      proc4.state = enqueue
1362    -> Input: 5.24 <-
1363      _process_selector_ = proc3
1364      proc4.running = FALSE
1365      proc3.running = TRUE
1366    -> State: 5.24 <-
1367      semaphore = 2
1368      proc3.state = idle
1369    -> Input: 5.25 <-
1370      _process_selector_ = proc0
1371      proc3.running = FALSE
1372      proc0.running = TRUE
1373    -> State: 5.25 <-
1374      semaphore = 3
1375      proc0.state = idle
1376    -> Input: 5.26 <-
1377    -> State: 5.26 <-
1378      proc0.state = enqueue
1379    -> Input: 5.27 <-
1380    -> State: 5.27 <-
1381      semaphore = 2
1382      apr = 1
1383      proc0.state = critical1
1384    -> Input: 5.28 <-
1385      _process_selector_ = proc1
```

```
1386      proc1.running = TRUE
1387      proc0.running = FALSE
1388    -> State: 5.28 <-
1389      semaphore = 3
1390      proc1.state = idle
1391    -> Input: 5.29 <-
1392    -> State: 5.29 <-
1393      proc1.state = enqueue
1394    -> Input: 5.30 <-
1395    -> State: 5.30 <-
1396      semaphore = 2
1397      apr = 2
1398      proc1.state = critical1
1399    -> Input: 5.31 <-
1400      _process_selector_ = proc2
1401      proc2.running = TRUE
1402      proc1.running = FALSE
1403    -> State: 5.31 <-
1404      semaphore = 3
1405      proc2.state = idle
1406    -> Input: 5.32 <-
1407    -> State: 5.32 <-
1408      proc2.state = enqueue
1409    -> Input: 5.33 <-
1410      _process_selector_ = proc3
1411      proc3.running = TRUE
1412      proc2.running = FALSE
1413    -> State: 5.33 <-
1414      proc3.state = enqueue
1415    -> Input: 5.34 <-
1416      _process_selector_ = proc2
1417      proc3.running = FALSE
1418      proc2.running = TRUE
1419    -> State: 5.34 <-
1420      semaphore = 2
1421      apr = 3
1422      proc2.state = critical1
1423    -> Input: 5.35 <-
1424      _process_selector_ = proc3
1425      proc3.running = TRUE
1426      proc2.running = FALSE
1427    -> State: 5.35 <-
1428      semaphore = 1
1429      apr = 4
1430      proc3.state = critical1
1431    -> Input: 5.36 <-
1432      _process_selector_ = proc4
1433      proc4.running = TRUE
1434      proc3.running = FALSE
1435    -> State: 5.36 <-
1436      semaphore = 0
1437      apr = 0
1438      proc4.state = critical1
1439    -> Input: 5.37 <-
1440    -> State: 5.37 <-
1441      proc4.state = critical2
1442    -> Input: 5.38 <-
1443    -> State: 5.38 <-
1444      proc4.state = exiting
1445    -> Input: 5.39 <-
```

```
1446   -> State: 5.39 <-
1447     semaphore = 1
1448     proc4.state = idle
1449   -> Input: 5.40 <-
1450     _process_selector_ = proc0
1451     proc4.running = FALSE
1452     proc0.running = TRUE
1453   -> State: 5.40 <-
1454     proc0.state = critical2
1455   -> Input: 5.41 <-
1456     _process_selector_ = proc1
1457     proc1.running = TRUE
1458     proc0.running = FALSE
1459   -> State: 5.41 <-
1460     proc1.state = critical2
1461   -> Input: 5.42 <-
1462     _process_selector_ = proc2
1463     proc2.running = TRUE
1464     proc1.running = FALSE
1465   -> State: 5.42 <-
1466     proc2.state = critical2
1467   -> Input: 5.43 <-
1468     _process_selector_ = proc3
1469     proc3.running = TRUE
1470     proc2.running = FALSE
1471   -> State: 5.43 <-
1472     proc3.state = critical2
```