

## Zadanie 1

```
lab14_1.smv U X
C: > Users > wiece > Desktop > Sem1 > MiASI > Lab14 > lab14_1.smv
1  MODULE main
2
3  VAR
4      cyfra : 0..9;
5      zamek: 0..4;
6      otwarty: boolean;
7
8  ASSIGN
9      init(cyfra) := 0..9;
10     init(zamek) := 0;
11     init(otwarty) := FALSE;
12
13     next(cyfra) := 0..9;
14
15     next(zamek) :=
16         case
17             zamek != 4 & next(cyfra) = 1
18             : 1;
19
20             zamek = 1 & next(cyfra) = 2
21             : 2;
22
23             zamek = 2 & next(cyfra) = 3
24             : 3;
25
26             zamek = 3 & next(cyfra) = 4
27             : 4;
28
29             zamek = 4
30             : 4;
31
32             TRUE
33             : 0;
34         esac;
35
36     next(otwarty) :=
37         case
38             zamek = 4
39             : TRUE;
40
41             TRUE
42             : FALSE;
43         esac;
44
```

## Zadanie 2

```

44
45 --Czy istnieje stan w którym zamek jest otwarty?
46 CTLSPEC EF(otwarty)
47
48 --Czy zawsze kiedy stan zamek jest różny od 4 to zamek jest zamknięty?
49 CTLSPEC AG (zamek != 4 -> otwarty = FALSE)
50
51 --Czy po wystąpieniu cyfry 1 i stanu zamku różnego od 4 zamek przejdzie w stan 1?
52 LTLSPEC G (next(cyfra) = 1 & zamek != 4 -> X zamek = 1)
53
54 --Czy po wystąpieniu cyfry różnej od 1 i wartości zamek równej 0 zamek przejdzie w stan 0?
55 LTLSPEC G (next(cyfra) != 1 & zamek = 0 -> X zamek = 0)
56
57 --Czy po wystąpieniu cyfry różnej od 1 i 2 i wartości zamek równej 1 zamek przejdzie w stan 0?
58 LTLSPEC G (next(cyfra) != 2 & next(cyfra) != 1 & zamek = 1 -> X zamek = 0)
59
60 --Czy po wystąpieniu cyfry różnej od 1 i 3 i wartości zamek równej 2 zamek przejdzie w stan 0?
61 LTLSPEC G (next(cyfra) != 3 & next(cyfra) != 1 & zamek = 2 -> X zamek = 0)
62
63 --Czy po wystąpieniu cyfry różnej od 1 i 4 i wartości zamek równej 3 zamek przejdzie w stan 0?
64 LTLSPEC G (next(cyfra) != 4 & next(cyfra) != 1 & zamek = 3 -> X zamek = 0)
65
66 --Czy zawsze po wystąpieniu cyfry 1 i zamku w stanie 0 zamek przejdzie do stanu 1?
67 LTLSPEC G (next(cyfra) = 1 & zamek = 0 -> X zamek = 1)
68
69 --Czy zawsze po wystąpieniu cyfry 2 i zamku w stanie 1 zamek przejdzie do stanu 2?
70 LTLSPEC G (next(cyfra) = 2 & zamek = 1 -> X zamek = 2)
71
72 --Czy zawsze po wystąpieniu cyfry 3 i zamku w stanie 2 zamek przejdzie do stanu 3?
73 LTLSPEC G (next(cyfra) = 3 & zamek = 2 -> X zamek = 3)
74
75 --Czy zawsze po wystąpieniu cyfry 4 i zamku w stanie 3 zamek przejdzie do stanu 4?
76 LTLSPEC G (next(cyfra) = 4 & zamek = 3 -> X zamek = 4)
77
78 --Czy po przejściu zamku w stan 4 zamek zostanie otwarty?
79 LTLSPEC G ((zamek = 4 -> X otwarty = TRUE))

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

PS C:\Users\wiece\Desktop\Sem1\WiAST\Lab14> .\NuSMV-2.6.0-win64\bin\NuSMV.exe lab14_1.smv
*** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:00 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 *** This version of NuSMV is linked to the zchaff SAT solver (see http://www.princeton.edu/~chaff/zchaff.html). ***
WARNING *** Zchaff is used in Bounded Model Checking when the system variable "sat_solver" is set to "zchaff". ***
WARNING *** Notice that zchaff is for non-commercial purposes only. ***
WARNING *** NO COMMERCIAL USE OF ZCHAFF IS ALLOWED WITHOUT WRITTEN PERMISSION FROM PRINCETON UNIVERSITY. ***
WARNING *** Please contact Sharad Malik (malik@ee.princeton.edu) for details. ***

```

```

-- specification EF otwarty is true
-- specification AG (zamek != 4 -> otwarty = FALSE) is true
-- specification G ((next(cyfra) = 1 & zamek != 4) -> X zamek = 1) is true
-- specification G ((next(cyfra) != 1 & zamek = 0) -> X zamek = 0) is true
-- specification G (((next(cyfra) != 2 & next(cyfra) != 1) & zamek = 1) -> X zamek = 0) is true
-- specification G (((next(cyfra) != 3 & next(cyfra) != 1) & zamek = 2) -> X zamek = 0) is true
-- specification G (((next(cyfra) != 4 & next(cyfra) != 1) & zamek = 3) -> X zamek = 0) is true
-- specification G ((next(cyfra) = 1 & zamek = 0) -> X zamek = 1) is true
-- specification G ((next(cyfra) = 2 & zamek = 1) -> X zamek = 2) is true
-- specification G ((next(cyfra) = 3 & zamek = 2) -> X zamek = 3) is true
-- specification G ((next(cyfra) = 4 & zamek = 3) -> X zamek = 4) is true
-- specification G (zamek = 4 -> X otwarty = TRUE) is true
PS C:\Users\wiece\Desktop\Sem1\WiAST\Lab14>

```

## Zadanie 3

```

lab14_1.smv U  lab14_3.smv U X
C: > Users > wiece > Desktop > Sem1 > MiASI > Lab14 > lab14_3.smv
1  MODULE  main
2
3  VAR
4      przycisk: Przycisk(piesi);
5      piesi: Swiatla2(samochody);
6      samochody: Swiatla3(przycisk);
7
8  DEFINE s := samochody.kolor;
9  DEFINE pi := piesi.kolor;
10 DEFINE pr := przycisk.zmien;
11
12 --Gdy światło dla pieszych jest zielone światło dla samochodów jest czerwone.
13 INVARSPEC pi = ziel -> s = czer
14

```

```

29  MODULE Przycisk(piesi)
30
31  VAR
32      zmien: boolean;
33      rand: 0..3;
34      countdown: 0..5;
35      wcisniety: boolean;
36
37  ASSIGN
38      init(zmien) := FALSE;
39      init(countdown) := 0;
40
41      next(zmien) :=
42          case
43              piesi.kolor = ziel
44              : FALSE;
45
46              countdown = 1
47              : TRUE;
48
49              zmien
50              : TRUE;
51
52              TRUE
53              : FALSE;
54          esac;
55
56      next(countdown) :=
57          case
58              countdown > 1
59              : countdown - 1;
60
61              zmien
62              : 0;
63
64              wcisniety
65              : 1..5;
66
67              TRUE
68              : 0;
69          esac;
70
71      next(wcisniety) :=
72          case
73              piesi.kolor = ziel
74              : FALSE;
75
76              TRUE
77              : rand = 0;
78          esac;
79

```

```

87
88  MODULE  Swiatla3(przycisk)
89
90  VAR
91      kolor: {ziel, zol, czer, czerZol};
92      countdown: 0..60;
93
94  ASSIGN
95      init(kolor) := ziel;
96      init(countdown) := 60;
97
98      next(kolor) :=
99          case
100              countdown > 1
101                  : kolor;
102
103              countdown = 0
104                  : kolor;
105
106              kolor = ziel & przycisk.zmien
107                  : zol;
108
109              kolor = zol
110                  : czer;
111
112              kolor = czer
113                  : czerZol;
114
115              kolor = czerZol
116                  : ziel;
117
118              TRUE
119                  : kolor;
120          esac;
121
122      next(countdown) :=
123          case
124              kolor = ziel & countdown = 1
125                  : 1;
126
127              countdown > 0
128                  : countdown - 1;
129
130              kolor = ziel
131                  : 60;
132
133              kolor = zol
134                  : 3;
135
136              kolor = czer
137                  : 18;
138
139              kolor = czerZol
140                  : 3;
141
142              TRUE
143                  : 0;
144          esac;
145

```

```

172
173
174
175  MODULE  Swiatla2(samochody)
176
177  VAR      kolor: {ziel, czer};
178          countdown: 0..15;
179
180  ASSIGN
181      init(kolor) := czer;
182      init(countdown) := 0;
183
184      next(kolor) :=
185          case
186              countdown > 1
187              : kolor;
188
189              countdown = 0
190              : kolor;
191
192              samochod.kolor != czer
193              : czer;
194
195              kolor = czer
196              : ziel;
197
198              kolor = ziel
199              : czer;
200          esac;
201
202      next(countdown) :=
203          case
204              countdown > 0
205              : countdown - 1;
206
207              kolor = czer & samochod.kolor = czer
208              : 1..2;
209
210              kolor = ziel
211              : 15;
212
213              TRUE
214              : 0;
215          esac;

```



Michał Więcek 248956

## Zadanie 4

```

lab14_1.smv U  lab14_3.smv U X
C: > Users > wiece > Desktop > Sem1 > MiASI > Lab14 > lab14_3.smv
1  MODULE  main
2
3  VAR
4      przycisk: Przycisk(piesi);
5      piesi: Swiatla2(samochody);
6      samochody: Swiatla3(przycisk);
7
8  DEFINE  s := samochody.kolor;
9  DEFINE  pi := piesi.kolor;
10 DEFINE  pr := przycisk.zmien;
11
12 --Gdy światło dla pieszych jest zielone światło dla samochodów jest czerwone.
13 INVARSPEC pi = ziel -> s = czerw
14
15 --Najkrótszy czas między naciśnięciem przycisku a światłem zielonym dla pieszych.
16 COMPUTE MIN[pr = TRUE, pi = ziel];
17
18 --Najdłuższy czas między naciśnięciem przycisku a światłem zielonym dla pieszych.
19 COMPUTE MAX[pr = TRUE, pi = ziel];
20
21 --Czy żółte światło dla samochodów nie wystąpi w przedziale od 0 do 59?
22 CTLSPEC !EBF 0..59 s = zol
23
24 --Czy czerwone światło dla samochodów jest zawsze w przedziale od 0 do 59?
25 CTLSPEC ABF 0..59 pi = czerw
26
27
28
29 MODULE  Przycisk(piesi)
30
31 > VAR      ...
32
33 > ASSIGN   ...
34
35 --Czy wciśnięty kiedyś będzie true?
36 CTLSPEC EF(wcisniety)
37
38 --Czy zawsze dopóki zmienna zmien jest równa false, zmienna countdown jest równa zero?
39 CTLSPEC A[countdown = 0 U zmien = FALSE]
40
41

```



```

85
86
87
88 MODULE Swiatla3(przycisk)
89
90 > VAR ...
94 > ASSIGN ...
146 --Czy jeśli przycisk.zmien jest true i świeci się zielone światło dla samochodów i minął licznik to kolor świateł zmieni się na żółty?
147 LTLSPEC G ((przycisk.zmien & kolor = ziel & countdown = 1) -> X kolor = zol)
148
149 --Czy światło samochodów będzie kiedyś zielone?
150 CTSPEC EF kolor = ziel;
151
152 --Czy światło samochodów będzie kiedyś żółte?
153 CTSPEC EF kolor = zol;
154
155 -- Czy światło samochodów będzie kiedyś czerwone?
156 CTSPEC EF kolor = czer;
157
158 --Czy światło będzie kiedyś czerwone i żółte jednocześnie?
159 CTSPEC EF kolor = czerZol;
160
161 --Czy zawsze jeśli kolor czerwono-żółty i minął licznik to w następnej klatce zapali się zielone?
162 LTLSPEC G (kolor = czerZol & countdown = 1 -> X kolor = ziel);
163
164 --Czy zawsze jeśli kolor czerwony i minął licznik to w następnej klatce zapali się czerwone i żółte jednocześnie?
165 LTLSPEC G (kolor = czer & countdown = 1 -> X kolor = czerZol);
166
167 --Czy zawsze jeśli kolor żółty i minął licznik to w następnej klatce zapali się czerwone?
168 LTLSPEC G (kolor = zol & countdown = 1 -> X kolor = czer);
169
170 --Czy zawsze jeśli kolor zielony i minął licznik istnieje możliwość by wrócił do światła żółtego?
171 CTSPEC AG (kolor = ziel & countdown = 1 -> EF kolor = zol);
172
173

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\wiece\Desktop\Sem1\MiASI\Lab14> .\NuSMV-2.6.0-win64\bin\NuSMV.exe lab14\_3.smv

\*\*\* This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:36:00 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 \*\*\* This version of NuSMV is linked to the zchaff SAT solver (see http://www.princeton.edu/~chaff/zchaff.html).

WARNING \*\*\* Zchaff is used in Bounded Model Checking when the system variable "sat\_solver" is set to "zchaff".

WARNING \*\*\* Notice that zchaff is for non-commercial purposes only.

WARNING \*\*\* NO COMMERCIAL USE OF ZCHAFF IS ALLOWED WITHOUT WRITTEN PERMISSION FROM PRINCETON UNIVERSITY.

WARNING \*\*\* Please contact Sharad Malik (malik@ee.princeton.edu) for details.

WARNING \*\*\* for details.

-- specification EF wcisniety IN przycisk is true

-- specification A [ countdown = 0 U zmien = FALSE ] IN przycisk is true

-- specification EF kolor = ziel IN samochody is true

-- specification EF kolor = zol IN samochody is true

-- specification EF kolor = czer IN samochody is true

-- specification EF kolor = czerZol IN samochody is true

-- specification AG ((kolor = ziel & countdown = 1) -> EF kolor = zol) IN samochody is true

-- specification !(EBF 0.59 s = zol) is true

-- specification ABF 0.59 pi = czer is true

-- the result of MIN [ pr = TRUE , pi = ziel ] is 0

-- the result of MAX [ pr = TRUE , pi = ziel ] is 67

-- specification G (((przycisk.zmien & kolor = ziel) & countdown = 1) -> X kolor = zol) IN samochody is true

-- specification G ((kolor = czerZol & countdown = 1) -> X kolor = ziel) IN samochody is true

-- specification G ((kolor = czer & countdown = 1) -> X kolor = czerZol) IN samochody is true

-- specification G ((kolor = zol & countdown = 1) -> X kolor = czer) IN samochody is true

-- invariant (pi = ziel -> s = czer) is true

PS C:\Users\wiece\Desktop\Sem1\MiASI\Lab14> █