Mateusz Markowski 235605

MiASI laboratorium 14

Konstrukcja i weryfikacja zsynchronizowanych automatów NuSMV
Prowadzący: dr inż. Paweł Głuchowski
Termin zajęć: Pn 17:05

Zad1.

```
MODULE main
VAR number : 0..9;
        state : {s0,s1,s2,s3,s4};
        open : boolean;
INIT state = s0 &
        open = FALSE;
TRANS next(number) in 0..9;
TRANS open in case
   state = s4 : TRUE;
   state in {s0,s1,s2,s3}: FALSE;
   esac;
TRANS next(state) in case
   (state = s0 & next(number)!=1) : s0;
   (state = s0 & next(number)=1) : s1;
   (state = s1 & next(number)=1) : s1;
   (state = s1 & next(number) in {0,3,4,5,6,7,8,9}) : s0;
   (state = s1 & next(number)=2) : s2;
   (state = s2 & next(number)=1) : s1;
   (state = s2 & next(number) in {0,2,4,5,6,7,8,9}) : s0;
   (state = s2 & next(number)=3) : s3;
   (state = s3 & next(number)=1) : s1;
   (state = s3 & next(number) in {0,2,3,5,6,7,8,9}) : s0;
   (state = s3 & next(number)=4) : s4;
   TRUE :s0;
   esac;
```

```
-- Czy number jest zawsze w zakresie 0-9
CTLSPEC AG(number in 0..9) -- true

-- Czy zamek jest zawsze zamknięty w stanach s0, s1, s2, s3
CTLSPEC AG(state in {s0, s1, s2, s3} -> open = FALSE) -- true

-- Czy zamek zawsze otwiera się w stanie s4
CTLSPEC AG(state = s4 -> open = TRUE) -- true

-- Czy zamek może zostać otwarty
CTLSPEC EF(open = TRUE) -- true

-- Czy sekwencja 1234 otwiera zamek?
CTLSPEC AG(AG(AG(AG(state = s0 & number = 1 -> state = s1) & number = 2 -> state =
s2) & number = 3 -> state = s3) & number = 4 -> state = s4 & open) -- true
```

Zad 2.

```
MODULE Swiatla3(t_2, kolor_2, wcisniety)
VAR t : 0..999;
        kolor : {zielone, zolte, czerwone, czerwone_zolte};
INIT t = 0 &
        kolor = zielone;
TRANS next(t) in case
   kolor=zielone & next(kolor)=zolte : 0;
   kolor=zolte & next(kolor)=czerwone : 0;
   kolor=czerwone & next(kolor)=czerwone_zolte : 0;
   kolor=czerwone_zolte & next(kolor)=zielone : 0;
   TRUE: (t+1);
   esac;
TRANS next(kolor) in case
   kolor=zielone & t>=60 & wcisniety : zolte; --zielone swiatlo zmienia sie na zolte, gdy
przycisk rozpocznie zmiane swiatel, ale nie wczesniej niż 60 sekund po jego zapaleniu
   kolor=zolte & t=3 : czerwone; --czas swiecenia zoltego swiatla (samego lub razem z
czerwonym) to 3 sekundy
   kolor=czerwone & kolor_2 = czerwone & t_2>=1 & t_2<2 : {czerwone,czerwone_zolte};
--losowosc 1-2 sekundy
   kolor=czerwone & kolor_2 = czerwone & t_2=2 : czerwone_zolte; --czerwone swiatlo
zmienia sie na czerwone z zoltym po 1-2 sekundach od zapalenia sie czerwonego swiatla na
sygnalizatorze dla pieszych
   kolor=czerwone_zolte & t=3 : zielone; --czas swiecenia zoltego swiatla (samego lub
razem z czerwonym) to 3 sekundy
   TRUE: kolor;
   esac;
```

```
MODULE Swiatla2(t_3, kolor_3)
VAR t : 0..999;
        kolor : {zielone, czerwone};
INIT t = 0 &
        kolor = czerwone;
TRANS next(t) in case
   kolor=zielone & next(kolor)=czerwone : 0;
   kolor=czerwone & next(kolor)=zielone : 0;
   TRUE: (t+1);
   esac;
TRANS next(kolor) in case
   kolor=czerwone & kolor_3 = czerwone & t_3>=1 & t_3<2 : {czerwone,zielone}; --losowosc
1-2 sekundy
   kolor=czerwone & kolor_3 = czerwone & t_3=2 : zielone; --czerwone swiatlo zmienia sie
na zielone po 1-2 sekundach od zapalenia sie czerwonego swiatla na sygnalizatorze dla
pojazdow
   kolor=zielone & t=15 : czerwone; --czas swiecenia zielonego swiatla to 15 sekund
   TRUE: kolor;
   esac;

MODULE Przycisk(kolor_2)
VAR wcisniety : boolean;
        wcisniety_wyslij : boolean;
        t : 0..999;
INIT t = 0 &
        wcisniety = FALSE &
        wcisniety_wyslij = FALSE;
TRANS next(t) in case
   wcisniety=FALSE & next(wcisniety)=TRUE : 0;
   wcisniety=TRUE & next(wcisniety)=FALSE : 0;
   TRUE: (t+1);
   esac;
TRANS next(wcisniety) in case
   wcisniety=FALSE & kolor_2 = czerwone : {TRUE, FALSE};--niewcisniety przycisk moze
zostac wcisniety w dowolnym momencie (losowo), kiedy sygnalizator dla pieszych swieci sie
na czerwono
   wcisniety=TRUE & next(kolor_2) = zielone : FALSE;--wcisniety przycisk zozostaje w tym
stanie az do zmiany sygnalizatora dla pieszych z zielonego swiatla na czerwone
   TRUE : wcisniety;
   esac;
TRANS next(wcisniety_wyslij) in case
   --wcisniecie przycisku rozpoczyna proces zmiany swiatel w obu sygnalizatorach po 1-5
sekundach.
   wcisniety = TRUE & t<1 : FALSE;
   wcisniety = TRUE & t>=1 & t<5 : {FALSE, TRUE};
   wcisniety = TRUE & t=5 : TRUE;
   TRUE : wcisniety_wyslij;
   esac;
```

```
MODULE main
VAR swiatla3 : Swiatla3(swiatla2.t, swiatla2.kolor, przycisk.wcisniety_wyslij);
        swiatla2 : Swiatla2(swiatla3.t, swiatla3.kolor);
        przycisk : Przycisk(swiatla2.kolor);



--sygnalizator dla pojazdow swieci sie na zielono tylko wtedy, gdy sygnalizator dla pieszych swieci sie na czerwono
CTLSPEC AG(swiatla2.kolor = zielone -> AX(swiatla3.kolor = czerwone)) --true

--sygnalizator dla pieszych swieci sie na zielono tylko wtedy, gdy sygnalizator dla pojazdow swieci sie na czerwono
CTLSPEC AG(swiatla3.kolor = zielone -> AX(swiatla2.kolor = czerwone)) --true

--czy reakcja na nacisniecie przycisku nastepuje po okreslonym czasie
CTLSPEC AG(przycisk.wcisniety = TRUE & przycisk.t < 1 & przycisk.t > 5 ->
AX(przycisk.wcisniety_wyslij = FALSE)) --true

--czy każde swiatlo kazdego sygnalizatora trwa tyle, ile trzeba
--wszystkie powinny byc false, ale sa true
CTLSPEC EF((swiatla3.t < 3 | swiatla3.t > 3) -> AX(swiatla3.kolor = czerwone))
CTLSPEC EF((swiatla3.t < 3 | swiatla3.t > 3) -> AX(swiatla3.kolor = zielone))
CTLSPEC EF(swiatla3.t < 60 -> AX(swiatla3.kolor = zolte))
CTLSPEC EF((swiatla2.t < 15 | swiatla2.t > 15) -> AX(swiatla2.kolor = czerwone))

--czy swiatla kazdego sygnalizatora swieca sie w dobrej kolejnosci
CTLSPEC A[TRUE U swiatla3.kolor = zielone -> AX(swiatla3.kolor = zolte)] --zamienione AF
--true
CTLSPEC AF(swiatla3.kolor = zolte -> AX(swiatla3.kolor = czerwone)) --true
CTLSPEC AF(swiatla3.kolor = czerwone -> AX(swiatla3.kolor = czerwone_zolte)) --true
CTLSPEC AF(swiatla3.kolor = czerwone_zolte -> AX(swiatla3.kolor = zielone)) --true

CTLSPEC AF(swiatla2.kolor = czerwone -> AX(swiatla2.kolor = zielone)) --true
CTLSPEC AF(swiatla2.kolor = zielone -> AX(swiatla2.kolor = czerwone)) --true

--minimalna sciezka miedzy zielonym, a czerwono zielonym swiatlem
COMPUTE MIN[swiatla3.kolor = zielone, swiatla3.kolor = czerwone_zolte] --25

--czy zawsze ktores swiatla sa czerwone
INVARSPEC (swiatla2.kolor = czerwone | swiatla3.kolor = czerwone) --true

--czy istnieje sciezka dlugosci 25 gdzie swiatlo jest zielone
CTLSPEC EBF 25..25 (swiatla3.kolor = zielone) --true
```