

MiASI laboratorium 15

Konstrukcja i weryfikacja automatów NuSMV do analizy programu

Prowadzący: dr inż. Paweł Głuchowski

Termin zajęć: Pn 17:05

Zad1.

MODULE main

FROZENVAR

number : 0..8;

VAR

e : {e1,e2,e3,e4,e5,e6,e7,e8,e9,e10,e11,e12};

i : 1..8;

factorial : 0..720; --1, 2, 6, 24, 120, 720

sigma : 0..5;

primes : 0..4; --{2, 3, 5, 7}

j : 3..9; --iterator petli zewnętrznej (j <= number)

x : 0..6; --iterator petli wewnętrznej (x < j - 2)

c : 0..6; --licznik wykonan pierwszej petli (j[3;9])

a : 1..3; --licznik deadlock

--sterowanie e

ASSIGN

init(e) := e1;

next(e) := case

e = e1 & number > 2 : e2;

e = e1 & number <= 2 : e12;

e = e2 & j <= number : e3;

e = e2 & j > number : e8;

e = e3 : e4;

e = e4 & x < (j - 2) : e5;

e = e4 & x >= (j - 2) : e7;

e = e5 : e6;

e = e6 : e4;

```

e = e7          : e2;
e = e8          : e9;
e = e9 & number = 3 : e10;
e = e9 & number != 3 : e11;
TRUE           : e;
esac;

```

--sterowanie i

ASSIGN

```

init(i) := 1;
next(i) := case
next(e) = e3          : 1;
next(e) = e6 & i < 8   : i + 1;
TRUE                 : i;
esac;

```

--sterowanie factorial

ASSIGN

```

init(factorial)      := 1;
next(factorial)      := case
next(e) = e3          : 1;
next(e) = e5 & (factorial * i <= 720) : factorial * i;
TRUE                 : factorial;
esac;

```

--sterowanie sigma

ASSIGN

```

init(sigma) := 0;
next(sigma) := case
next(e) = e7 & (sigma + (factorial - j * (factorial / j)) <= 5) : sigma + (factorial - j *
(factorial / j));
TRUE                                     : sigma;
esac;

```

--sterowanie primes

ASSIGN

```

init(primes) := 0;
next(primes) := case
next(number) = 3 : 2;
next(e) = e8 & (sigma - 1) <= 4 & (sigma - 1) >= 0 : sigma - 1;
TRUE                                     : primes;
esac;

```

--sterowanie j

ASSIGN

```

init(j) := 3;
next(j) := case
e = e1 & next(e) = e2      : j;

```

```

e = e7 & next(e) = e2 & j < 9 : j + 1;
TRUE                          : j;
esac;

```

--sterowanie x

ASSIGN

```

init(x) := 0;
next(x) := case
e = e3 & next(e) = e4      : 0;
e = e6 & next(e) = e4 & x < 6 : x + 1;
TRUE                      : x;
esac;

```

--sterowanie c

ASSIGN

```

init(c) := 0;
next(c) := case
(c + 1) > 6      : c;
next(e) = e3 & c < 6 : c + 1;
TRUE            : c;
esac;

```

--sterowanie a

ASSIGN

```

init(a) := 1;
next(a) := case
a = 1
& next(e) = e
& next(i) = i
& next(factorial) = factorial
& next(sigma) = sigma
& next(primes) = primes
& next(j) = j
& next(x) = x
& next(c) = c : 2;
a = 2          : 3;
TRUE          : a;
esac;

```

Zad 2.

-- czy podanie number w przedziale [0;2] prowadzi do wydrukowania napisu "wrong number"?

CTLSPEC AG(number in 0..2 -> AX(e = e12)) --true

-- SPRAWDZENIE LICZBY LICZB PIERWSZYCH BEZ MIN/MAX

-- jaka jest najmniejsza i największa liczba liczb pierwszych (primes) dla number [3;8]

-- dla [3;4] - 2; dla [5;6] - 3; dla [7;8] - 4

-- czy nieistnieje stan w którym number należy do przedziału [3;8] znajdujemy się w stanie końcowym i mamy X liczb pierwszych

--CTLSPEC !EF(number in 3..8 & a = 2 & primes > 4) -- true - nie istnieje taki stan dla których mamy więcej niż 4 liczby pierwsze

--CTLSPEC !EF(number in 3..8 & a = 2 & primes = 4) -- false - istnieje taki stan dla których mamy 4 liczby pierwsze

--CTLSPEC !EF(number in 3..8 & a = 2 & primes = 3) -- false - istnieje taki stan dla których mamy 3 liczby pierwsze

--CTLSPEC !EF(number in 3..8 & a = 2 & primes = 2) -- false - istnieje taki stan dla których mamy 2 liczby pierwsze

--CTLSPEC !EF(number in 3..8 & a = 2 & primes < 2) -- true - nie istnieje taki stan dla których mamy mniej niż 2 liczby pierwsze

-- najmniejsza liczba liczb pierwszych - 2

-- największa liczba liczb pierwszych - 4

-- jaka jest maksymalna długość ścieżki stanów do stanu końcowego gdy number = 8?

COMPUTE MAX[a = 1 & number = 8, a = 2] -- 92

--jaka jest największa liczba liczb pierwszych (odczytać ostatnią wartość primes z wyniku)

CTLSPEC !EBF 92..92(number = 8) -- 4

-- jaka jest minimalna długość ścieżki stanów do stanu końcowego gdy number = 3?

COMPUTE MIN[a = 1 & number = 3, a = 2] -- 1

--jaka jest najmniejsza liczba liczb pierwszych (odczytać ostatnią wartość primes z wyniku)

CTLSPEC !EBF 1..1(number = 3) -- 2

-- jaka jest maksymalna długość ścieżki stanów do stanu końcowego gdy number = 6?

COMPUTE MAX[a = 1 & number = 6, a = 2] -- 51

-- ile razy wykona się zewnętrzna pętla for dla number = 6 (odczytać ostatnią wartość c z wyniku)

CTLSPEC !EBF 51..51(a = 2) -- 4