

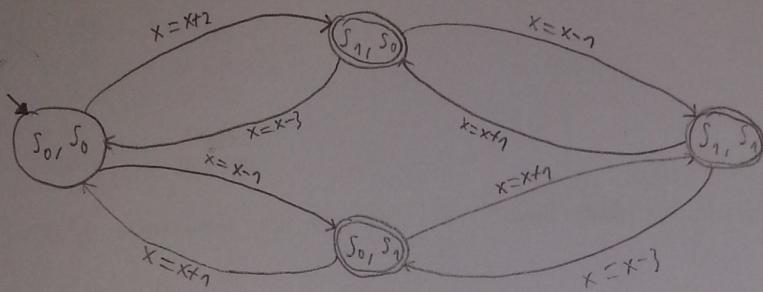
Predložak Promela programa za laboratorijsku vježbu ($N = 70$):

```
1  #define N 40
2  #define p  (x<=0)
3
4  int x = N;
5
6  active proctype A()
7  {
8      do      :: x=x-1 : x=x-1
9      od;
10
11 }
12
13 active proctype B()
14 {
15     do      :: x=x-1 : x=x-1
16     od
17
18 }
19
20 never { /* LTL formula */ /* <>p */}
21    T0-init;
22    do
23        :: atomic {((p))} -> next({!((p))})
24        :: (p) -> yolo T0-init
25
26 accept-all:
27     skip
28
29
30
31
32 }
```

a) FSA $A_1 = (\{s_0, s_1\}, S_0, \{x=x+2, y=x-3\}, \{(s_0, x=x+2, s_1), (s_1, x=x-3, s_0)\}, \{s_1\})$

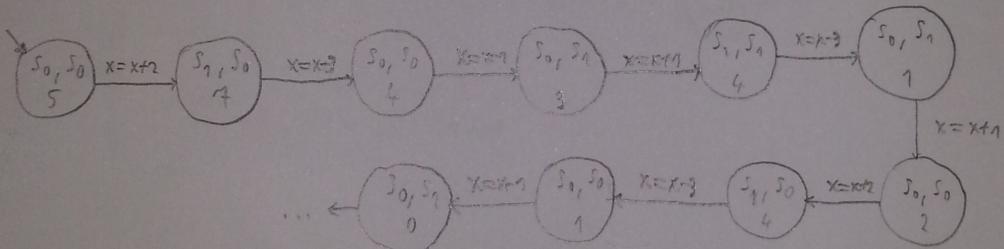
FSA $A_2 = (\{s_0, s_1\}, S_0, \{x=x-7, y=x+7\}, \{(s_0, x=x-7, s_1), (s_1, x=x+7, s_0)\}, \{s_1\})$

b)



c)

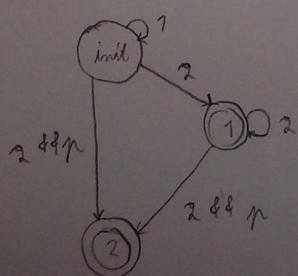
$x=5$



d) $\square p, \forall x \geq 0$ NIE ISTINA jer se stavlja s_1, s_0 i s_1, s_1 . MOGU BESKONACNO ponavljati pa za odredeni $x > 0$ to ne vrijedi.

e) $\square p, \forall x \geq 0$ NIE ISTINA jer se stavlja s_1, s_0 i s_1, s_1 mogu beskonacno ponavljati pa za odredeni $x > 0$ to ne vrijedi.

f)



3) $x = x + 2$ $x = 72$
 $x = x - 1$ $x = 71$
 $x = x + 1$ $x = 72$
 $x = x - 1$ $x = 71$
 $x = x - 3$ $x = 68$
 $x = x + 1$ $x = 69$
 $x = x - 1$ $x = 68$
 $x = x + 1$ $x = 69$
 $x = x + 2$ $x = 71$
 $x = x - 3$ $x = 68$
 $x = x - 1$ $x = 67$
 $x = x + 1$ $x = 68$

NEŠTA NEKE VECI RAZLIKU OSIM ŠTO JE POČETNA VRIJEDOST
 X DRUGACIJA I ŠTO REDOSLJED PRVELAZA NIJE ISTI, ALI
 TO JE OČEKIVANO.

6) UVJET JE ZADOVOLJEN, TO VIDIMO (2) "depth reached 556, errors: 1"

7) POSTOJI TAKVA SEKVENCA, 554 KORAKA.

8) Never { /* !<=2 */ }

accept-init:

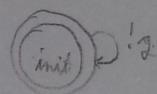
TO-init:

do

i: (!((2))) → goto TO-init

od:

}



9) Never { /* ![]_2 */ }

TO-init:

do

:: atomic { (!((2))) → never (!(!((2))!)) }

:: (1) → goto TO-init

od:

accept-all:

skip

}



10)

never { !# <> (p \vee q) * }

T0-init:

do

:: ((q)) \rightarrow goto accept_S4

:: atomic { ((p)) $\&$ (q)) \rightarrow assert (!((p) $\&$ (q))) }

:: (r) \rightarrow goto T0-init

od;

accept_S4:

do

:: ((q)) \rightarrow goto accept_S4

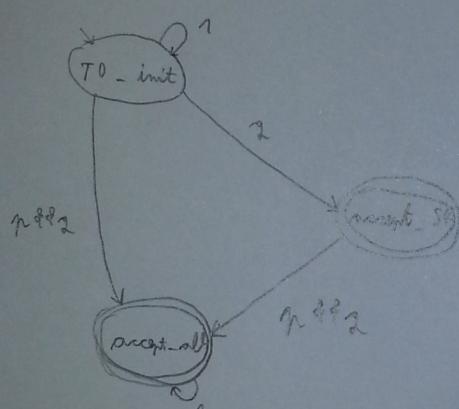
:: atomic { ((p)) $\&$ (q)) \rightarrow assert (!((p) $\&$ (q))) }

od;

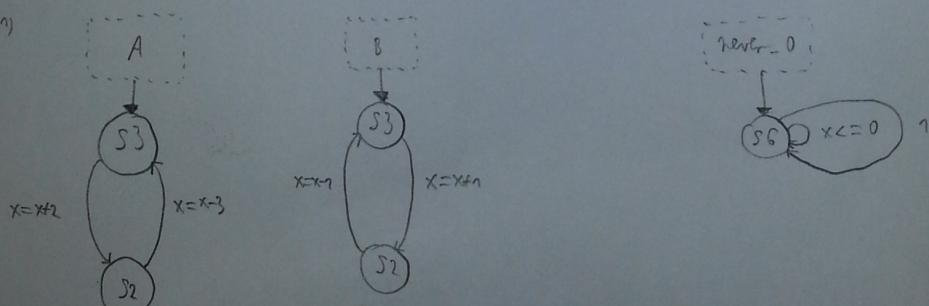
accept_all:

skip

}



11)



NIJE BILJNO NIJE RAZLICITO. SAMO SU NAZIVI STANJA DRUGACIJI I POSTOJI FSA ZA never BLOK.

p1) postoji.

p2) 560 koraka.

p3) NIJE MOGUC JER BI ASSERT TREBALO IZVODITI TEK NAKON ODRZENOG TRENUTKA OD KADA BI UNIJEZ TREBALO VRIVEDITI.

p4) NIJE MOGUC JER POSTOJI BESKONACNO MNOGO PUTOVA IZVODENJA.