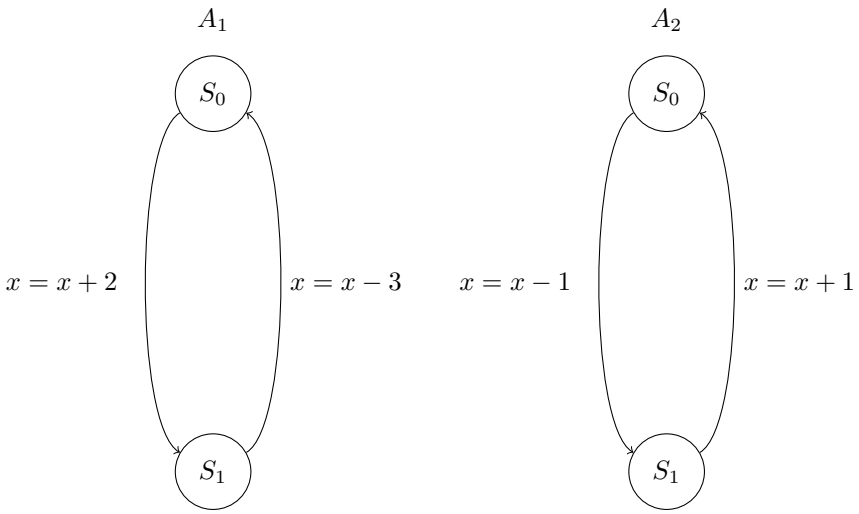
Formalne metode u oblikovanju sustava

Upute za 3. domaću zadaću
(8. 9 i 10. predavanje)

Kolokvij se piše u petak 8. lipnja 2018. godine u 18:00 sati u dvoranama B1, D1 i D2 godine prema rasporedu koji je objavljen na web stranicama predmeta. Studenti na kolokvij mogu donijeti vlastoručno napisanu domaću zadaću kako bi lakše položili kolokvij. Studenti koji žele mogu na posebnim papirima predati rješenja za zadatke označene zvjezdicom.

Ime i prezime: _____ Potpis: _____

Zadana su dva konačna diskretna automata A_1 i A_2 prema slici:
(početna stanja su uvijek S_0 a završna S_1)



Slika 1: FSA A_1 i A_2

U okviru domaće zadaće potrebno je:

a) Detaljno opisati strukturu automata A_1 i A_2 prema definiciji $FSA\ A = (S, s0, L, T, F)$ (odrediti elemente svakog od skupova S, s_0, L, ...)

b) Odrediti asinkroni produkt automata A_1 i A_2 i nacrtati ga.

c) Odrediti ekspandirani asinkroni produkt za prvih 5–10 po volji odabranih članova i nacrtati ga.

d) Pomoću ekspandiranog produkta odrediti istinitost LTL formule $\Diamond\Box p$ ako je $p \equiv x \leq 0$. Obrazložiti rješenje, posebice komentirati mogućnost rješavanja bez primjene programskih alata.

e) Pomoću ekspandiranog produkta odrediti istinitost LTL formule $\Diamond p$ ako je $p \equiv x < 0$. Obrazložiti rješenje, posebice komentirati mogućnost rješavanja bez primjene programskih alata.

f) Nacrtati moguću realizaciju *Büchi* automata za *LTL* formulu: $\Diamond(p \vee q)$.

1) Instalirajte programski alat Spin (http://spinroot.com/spin/Bin/index.html). Instalacije se svodi na kopiranje izvršnog programa.
Za one koji hoće više, sve instrukcije jezika *Promela* možete pronaći na http://spinroot.com/spin/Man/promela.html kao i službene upute ("manual") na http://spinroot.com/spin/Man/Manual.html).

2) Editirajte automate A_1 i A_2 kao promela procese `A` i B (vidjeti predložak na strani **6**). Napomena: *Promela* file nazvati `prezime.prm` (npr. `blaskovic.prm`). Polazeći od zadanog *Promela* modela koji se sastoji od dva procesa A i B analizirat će se `LTL` formula $\Diamond p$ gdje je $p \equiv (x <= 0)$.

3) Pokrenite simulaciju: `spin -u20 -p -c -g prezime.prm`.
Prepišite prvih 12 članova. Pismeno obrazložite istovjetnosti i razlike između ekspandiranog asinkronog produkta iz domaće zadaće i rezultata simulacije.

4) Generirajte analizator: `spin -a prezime.prm`.

5) Prevedite u izvršni oblik npr.: `gcc -o pan pan.c`.

6) Pozovite analizator: `pan -a` ili `./pan -e`.
Pismeno obrazložite da li je uvjet p zadovoljen ?

Dodatna napomena: isprobajte i naredbu `spin -run prezime.prm` koja zamjenjuje korake 4. - 6.

7) Pokrenite "error trail" opciju (pronalaženje protuprimjera) sa `spin -t -p -c -g prezime.prm`. Da li postoji sekvenca u kojoj varijabla x na kraju poprima vrijednost $x \leq 0$? Koliko koraka (*eng. "steps"*) sadrži ?

8) Prepišite instrukcije za Büchi automat koje generira Spin `spin -f '!<>q'`. Nacrtajte pripadni Büchi automat.

9) Prepišite instrukcije za Büchi automat koje generira Spin `spin -f '![]q'`. Nacrtajte pripadni Büchi automat.

10) Na isti način koristeći Spin nacrtajte automat iz Vaše domaće zadaće (pitanje `f)`).

11) Generirajte analizator sa `spin -a -o3 prezime.prm`, prevedite te pozovite analizator s `pan -d`.
Precrtajte tako dobivene *FSA*. Objasnite razlike kao i istovjetnosti prema automatima iz domaće zadaće ? Usporedite stanja prema slici na stranici 1 i stanja dobivena s opcijom `pan -d`. U čemu je razlika ?

Odgovorite na sljedeća pitanja:
Ponovite postupak za $\Diamond\Box p$ (modificirati "*never claim*" `spin -f '<>[]p'` na kraju `prezime.prm` datoteke.

p1) Da li postoji sekvenca u kojoj varijabla x na kraju poprima vrijednost $x \leq 0$?

p2) Koliko koraka (*eng. "steps"*) sadrži ?

p3) Da li je moguće problem riješiti bez *LTL* formule samo pomoću *assert* naredbi ? Obrazložite odgovor.

p4) Da li je moguće problem riješiti bez *LTL* formule samo pomoću simulacije ? Obrazložite odgovor.

Zadan je Promela model komunikacijskog protokola koji opisuje dio moguće realizacije protokola za preuzimanje datoteka (*eng. "download"*). Ako je $N = 24$, pomoću programskog alata `Spin` odredite:

a) stanja iz kojih nema napretka (*eng. "deadlock"*),

b) dolazi li protokol u završno stanje ?

Ako je potrebno odredite protuprimjere.

⋆Obrazložite da li je potrebno uvoditi dodatne instrukcije pomoću *LTL* formule koje će omogućiti provjeru postojanja stanja iz kojih nema napretka (*eng. "deadlock"*) ? (nije obavezno pitanje)

```
1
2
3  mtype = { ini, ack, dreq, data, shutup, quiet, dead };
4
5  #define N
6
7  chan M = [N] of { mtype };
8  chan W = [N] of { mtype };
9
10 active proctype Mproc()
11 {
12     W!ini;                  /* connection      */
13     M?ack;                  /* handshake       */
14
15     timeout ->              /* wait            */
16     if                      /* two options:    */
17     :: W!shutup             /* start shutdown  */
18     :: W!dreq;              /* or request data */
19         M?data ->           /* receive data    */
20         do
21         :: W!data           /* send data       */
22         :: W!shutup;        /* or shutdown     */
23             break
24         od
25     fi;
26
27     M?shutup;               /* shutdown handshake */
28     W!quiet;
29     M?dead
30 }
31
32 active proctype Wproc()
```

```
33 {
34     W?ini;                  /* wait for ini    */
35     M!ack;                  /* acknowledge     */
36
37     do                      /* 3 options:      */
38     :: W?dreq ->            /* data requested  */
39         M!data              /* send data       */
40     :: W?data ->            /* receive data    */
41 #if 1
42         M!data
43 #else
44         skip                /* no response     */
45 #endif
46     :: W?shutup ->
47         M!shutup;           /* start shutdown  */
48         break
49     od;
50
51     W?quiet;
52     M!dead
53 }
54
```

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

```
#define N

#define p

int x = N;

active proctype A()
{
do
        ..............
od;
}

active proctype B()
{
do
        ...........
od
}

never {    /* LTL formula */









}

```

($\star$) Rješenja za ovaj zadatak uz detaljna obrazloženja možete priložiti uz kolokvij (nije obavezno).

(nIQt problem poznat iz AI)

Na obali rijeke nalaze se *tata*, *mama*, dvije kćeri (*kci1* i *kci2*), dva sina (*sin1* i *sin2*), zatvorenik i stražar (*zatv* i *straz*). Ako splav koja će ih prebaciti na drugu stranu rijeke može nositi samo dvije osobe potrebno je definirati `Promela` model koji će pronaći raspored (sekvencu ili slijed akcija) kojom će svi (tata, mama, sinovi, kćeri prijeći na drugu stranu rijeke poštujući slijedeća ograničenja – pravila:

(1) najviše dvije osobe mogu istovremeno biti na splavi

(2) *tata* ne smije biti s kćerima bez prisustva *mame*

(3) *mama* ne smije biti sa sinovima bez prisustva *tate*

(4) zatvorenik ne smije biti s bilo kojim članom obitelji bez prisustva stražara

(5) jedino roditelji (*tata* i *mama*) te stražar (*straz*) mogu upravljati sa splavi

Kod rješavanja kao varijablu za trenutni položaj možete na primjer koristiti polje bitova $Pl[i]$, gdje je "i" *tata*, *mama* ...
Na polaznoj strani, $Pl[i]$ je neistinit, a na odredišnoj strani je $Pl[i]$ je istinit.
Kod rješavanja možete po slobodnom izboru koristiti model s ili bez protuprimjera, modelirati kao jedan ili više procesa, odnosno koristiti globalne varijable ili komunikaciju preko kanala.

($\star$) Zadana su dva procesa *inc()* i *dec()*. Potrebno je modificirati modele tako da oba procesa dođu u završno stanje . Pri tome je potrebno:

a) nadopuniti *LTL* formulu kojom se provjerava nepostojanje zastoja (eng. *deadlock*): formula mora biti istinita i ne smije doći do generiranja protuprimjera,

b) unijeti potrebne modifikacije u modele procesa,

c) nije dozvoljeno korištenje naredbe `timeout`.

```
int x = 0;
bool dinc = false;
bool ddec = false;

ltl LTL0 { [] (       ); }  // nadopuniti LTL formulu

proctype inc() {
   int t;
   atomic {
      t = x;
      x = t + 1;
   }
   dinc = true;
}

proctype dec(){
    int t;
    t = x;
    x = t - 1;
    ddec =   true;
    }

init { run inc();
       run dec();
     }
```

($\star$) Za one koji hoće više: ovdje možete uvesti vlastiti problem i predložiti rješenja pomoću modeliranju u jeziku *Promela* te analizom alatom *Spin*. Rješenja za ovaj zadatak uz detaljna obrazloženja možete priložiti uz kolokvij (nije obavezno).