Formalne metode u oblikovanju sustava
Test pitanje br. 7594i53

Ime i prezime: _____ Mat.br. _____

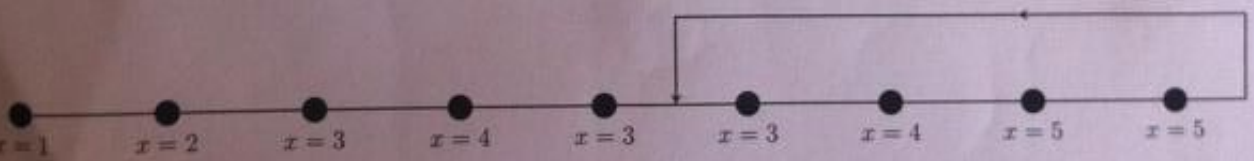
1) Opisati što radi naredba:

```
spin -p -c -g -l -u72 model.prml
```

2) Odredite: dolazi li proces *FSA* u završno stanje ili ostaje blokiran ? Koju vrijednost na kraju poprima varijabla *a*? Obrazložite odgovor ! Opišite kako se može osigurati da *promela* model uvijek dođe do naredbe u 10 redu (do regularnog završetka) i da pri tome uvijek $assert(a == 2)$ bude istinito ?

```
1  active [8] proctype  FSA() {
2    byte a=2;
3  if
4  :: (a==2) -> a--;
5  :: (a==1) -> a++;
6  :: else -> goto end_FSA;
7  fi
8  end_FSA:
9          printf("a=%d\n",a); assert(a==2);
10 }
```

3) Opišite što provjerava temporalna formula: $\Box\Diamond(x > 5)$
te odredite istinitost temporalne formule. Napomena: promatrati samo sekvencu σ (dio ekspandiranog produkta) prema slici.



4) Skicirajte moguću realizaciju Büchi automata za slijedeću *LTL* formulu:
$\Diamond\Box p$

6) * Zadan je poslovni proces – aplikacija "Odobravanje kredita" (**LoanApproval**) koga je potrebno modelirati `promela` procesima. "Odobravanje kredita" sastoji se od tri manja procesa:

(P1) *Kupac* (**"Customer"**)

(P2) *KreditniReferent* (**"Loan-Approver"**)

(P3) *ProcjeniteljRizika* (**"RiskAssessor"**)

Razmjena poruka:

(1) Najprije *Kupac* (**"Customer"**) šalje poruku sa zahtjevom "request–small" (traži se mali iznos kredita) ili "request–large" (traži se veliki iznos kredita) u proces *KreditniReferent* (**"Loan-Approver"**).

(2) Ako je iznos kredita mali *KreditniReferent* (**"Loan-Approver"**) šalje potvrdu o odobrenju kredita (`approved`) procesu *Kupac* (**"Customer"**) te završava aplikaciju.

(3) Ako se traži veliki iznos kredita tada *KreditniReferent* (**"Loan-Approver"**) šalje poruku u proces *ProcjeniteljRizika* (**"RiskAssessor"**) sa zahtjevom za procjenu.

(4) Nakon toga *ProcjeniteljRizika* (**"RiskAssessor"**) šalje poruku `risk--high` (visoki–rizik) ili `risk--low` (niski–rizik) u proces *KreditniReferent* (**"Loan-Approver"**)

(5) Ako *KreditniReferent* (**"Loan-Approver"**) primi poruku `risk--high` (visoki–rizik) tada šalje poruku o odbijanju kredita (`denied`) procesu *Kupac* (**"Customer"**).

(6) Ako *KreditniReferent* (**"Loan-Approver"**) primi poruku `risk--low` (niski–rizik) tada šalje poruku o prihvaćanju kredita (`approved`) procesu *Kupac* (**"Customer"**).

Valja odrediti:

a) `Promela` modele za svaki od procesa (*Kupac* ...)

b) za svaki od procesa nacrtati pripadne automate (FSA)

c) opisati postupak za određivanje zastoja (`''deadlock''`) i nedostupnog koda.

d) Napisati pripadne predikate i LTL formule za provjeru sljedećih svojstava:

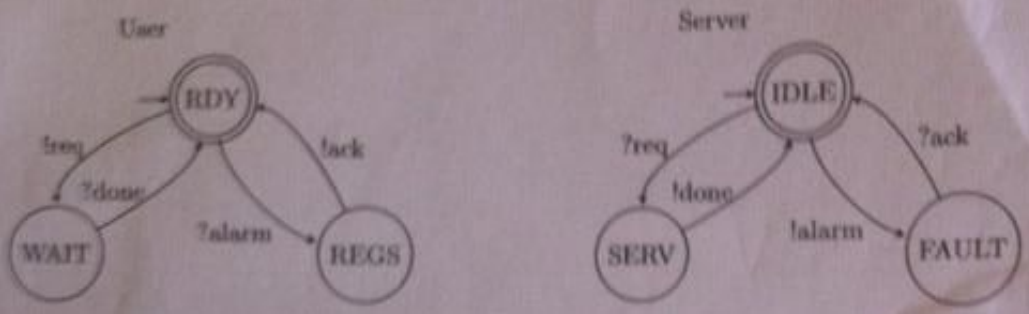
(p1) Kredit će eventualno biti ili prihvaćen ili odbijen

(p2) Ako je traženi iznos kredita mali kredit će eventualno biti prihvaćen

(p3) Ako je traženi iznos kredita velik kredit će eventualno biti ili prihvaćen ili odbijen

Komunikaciju među procesima modelirati sinkronim i asinkronim kanalima, po potrebi uvesti nove poruke ili varijable.

5) Na slici su prikazani konačni automati (FSA) koji opisuju komunikacijski sustav. Sustav se sastoji od dvije komponente: *User* i *Server*. Komponente izmjenjuju poruke asinkrono preko komunikacijske mreže koju predstavljamo komunikacijskim kanalima (*ch*) kapaciteta N.



Potrebno je:

a) specificirati pripadne `Promela` modele. Komunikaciju riješiti preko kanala poznatog kapaciteta.

b) Da li sustav može ostati u stanju iz kojeg nema napretka, tj. dolazi li do zastoja ? (eng. `''deadlock''`)

Detaljno predočite postupak za određivanje zastoja. Po vlastitom izboru koristite poznate `Promela` naredbe, LTL formule, naredbu `assert` ...

c) modificirajte `Promela` model tako da se sastoji od dvije istovrsne inačice komponente `User`. Detaljno obrazložite potrebne modifikacije. Ovako modificirani model nije potrebno dalje analizirati.

d) Koristeći naredbu `len(ch)`, definirajte postupak koji će provjeriti ako dolazi do "gubitka poruka". Do gubitka poruka dolazi: ako neki od `Promela` procesa pokušava zapisati poruku u "puni" komunikacijski kanal, tj. ako je kanal ch pun, tada je $(len(q) == N)$ istinito.

Originalni zapis iz uputstva za naredbu `len(q)`:

len - predefined, integer function to determine the number of messages that are stored in a buffered channel.

e) * Da li zatim definirani `Promela` model zadovoljava svojstvo životnosti (eng. `liveness`). Predložite i obrazložite rješenje.

f) * Usporedite komunikaciju procesa preko sinkronih i asinkronih kanala. Za oba slučaja obrazložite može li doći do zastoja.

Napomene: Ako smatrate potrebnim, slobodno uvedite parametre koji nisu precizirani u tekstu zadatka. Zadatke označene s * nije obavezno rješavati, predviđeni su "za one koji hoće više".