

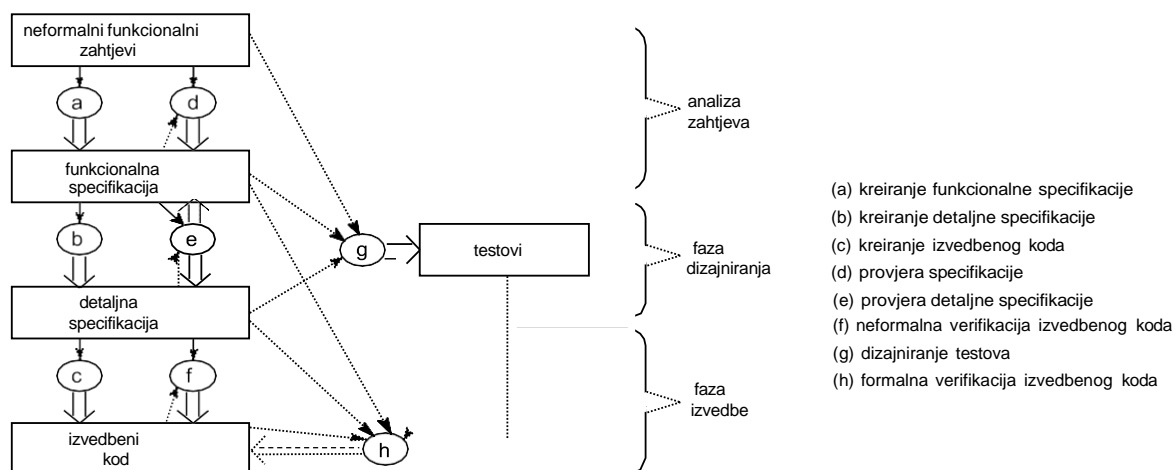
Laboratorijska vježba br. 2

PRIMJENA SPIN-a PRI PROVJERI LOGIČKE KONZISTENTNOSTI I VERIFIKACIJI MODELA PROTOKOLA, PETRIJEVE MREŽE I FORMALNE METODE SPECIFIKACIJE I SIMULACIJSKO MODELIRANJE PROTOKOLA

1. PRIMJENA SPIN-a PRI PROVJERI LOGIČKE KONZISTENTNOSTI I VERIFIKACIJI MODELA PROTOKOLA

U primjeni se nalazi veliki broj sustava kod kojih se procesi izvode na različitim međusobno udaljenim uređajima - *distribuirani sustavi*. U takvim sustavima procesi međusobno komuniciraju putem poruka koje se razmjenjuju odgovarajućim komunikacijskim protokolima. Ti protokoli definirani su ovisno o karakteristikama procesa i dostupnih prijenosnih puteva. Budući da se neprestano javlja potreba za međusobnim povezivanjem različitih procesa, stalna je i potreba za kreiranjem novih komunikacijskih protokola, specifičnih za svaku pojedinu primjenu.

Kreirati protokol specifičan za određenu primjenu složen je proces koji se provodi kroz nekoliko faza. Kako je prikazano slikom 1., u faze razvoja protokola ubrajaju se analiza zahtjeva, dizajniranje te izvedba protokola.



Slika 1. Faze razvoja protokola

Metode specifikacije protokola koje se primjenjuju temelje se na određenim modelima. Modeli se primjenjuju kako bi se pojednostavio prikaz i opis složenih pojava. Pri procesu razvoja modela bitno je primijeniti provjerene metode razvoja algoritama.

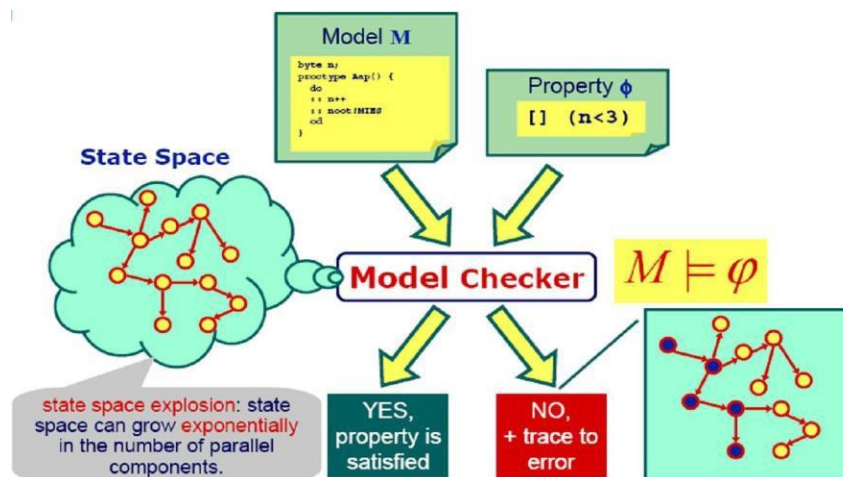
Usljed složenosti procesa dizajniranja, u svakoj od faza razvoja protokola može doći do pogreške. Pogreške nastale u fazi kodiranja najčešće su posljedica nekonzistentnog dizajna samog protokola. Zato je poželjno razviti metode otkrivanja pogrešaka u ranim fazama njihovog kreiranja.

Većina pogrešaka u dizajnu protokola može se otkriti pomoću određenih tehnika za provjeru modela sustava. Provjera modela je automatizirana tehnika koja za model sustava s konačnim brojem stanja i neko logičko svojstvo sustavno provjerava očuvanje tog svojstva za dani model.

Dva osnovna pristupa koja se primjenjuju pri provjeri modela su:

- **verifikacija** - utvrđivanje ispravnosti detaljnog modela sustava koji je u postupku validacije (verifikacija protokola vrši se prije nego što se priđe implementaciji protokola);
- **traženje pogrešaka** - traženje pogrešaka u modelu sustava.

Dakle, u postupku razvoja protokola nužno je izvršiti detaljnu verifikaciju kompletnosti i logičke konzistentnosti protokola.



Slika 2. Provjera modela

Na tržištu postoji niz alata koji se koriste za provjeru logičke konzistentnosti modela. Jedan od kvalitetnijih besplatnih alata namijenjenih verifikaciji i provjeri logičke konzistentnosti modela s konačnim brojem stanja - komunikacijskih protokola, je SPIN.

Preporučena literatura:

- prezentacije i bilješke s auditornih vježbi i predavanja
- www.spinroot.com
- G. J. Holzmann: "The Spin Model Checker: Primer and Reference Manual", Addison Wesley, 2003.
- G. J. Holzmann: "Design and Validation of Computer Protocols", Prentice Hall, 1990.

Priprema za vježbu:

Prije vježbi potrebno je proučiti predložak za vježbu te navedenu literaturu.

Cilj vježbe:

Cilj ove vježbe je primjena SPIN-a pri simulaciji i verifikaciji komunikacijskih protokola opisanih Promelom. Potrebno je razumjeti postupak te znati provjeriti valjanost jednostavnih protokola primjenom SPIN programa.

PRIMJERI I ZADACI

I.) IZVOĐENJE SINKRONIZIRANIH PROCESA

Zadatak 1:

Zadan je sljedeći program:

```
mtype = { msgtype };
chan name = [0] of { mtype, byte };
active proctype A()
{ name!msgtype(128);
  name!msgtype(125) }
active proctype B()
{ byte state;
  name?msgtype(state) }
```

- Promotrite drugu liniju u zadanom kodu. Koliki je kapacitet kanala? Što to znači za poruke koje stižu u taj kanal?
- Da li se radi o sinkronoj ili asinkronoj komunikaciji između procesa? Po čemu to zaključujete?
- Objasnite što radi ovaj program komentiranjem linija programa.
- Izvedite simulaciju te verifikaciju programa. Zašto dolazi do blokiranja, tj. zašto se ne može izvesti druga linija u procesu A?

U predviđene prostore upišite rješenja zadataka.

Rješenje:

a) Odgovor:

Kapacitet kanala je 0. Za sve poruke koje stižu u taj kanal znači da će se zadržavati samo trenutno.

b) Odgovor:

Kapacitet kanala je 0 što znači da je komunikacija sinkrona.

c) Odgovor:

Ime kanala je globalni port preko kojeg procesi uspostavljaju komunikaciju. Preko kanala `name!msgtype(128)` proces A šalje poruku, dok proces B prima tu istu poruku preko kanala `name?msgtype(state)`. Izjava u procesu A neće biti izvršena zato što ne postoji operacija primanja u B procesu.

d) Odgovor:

S obzirom da je u procesu B napisana samo jedna linija koda za primanje jedne varijable druga linija u procesu A se neće izvršiti.

II.) IZVOĐENJE NESINKRONIZIRANIH PROCESA

Zadatak 2:

Zadan je sljedeći program:

```
proctype A(chan q1)
{
  chan q2;
  q1?q2;
  q2!123
}
proctype B(chan qforb)
{
  int x;
  qforb?x;
  printf("x = %d\n", x)
}
init {
  chan qname = [1] of { chan };
  chan qforb = [1] of { int };
  run A(qname);
  run B(qforb);
  qname!qforb
}
```

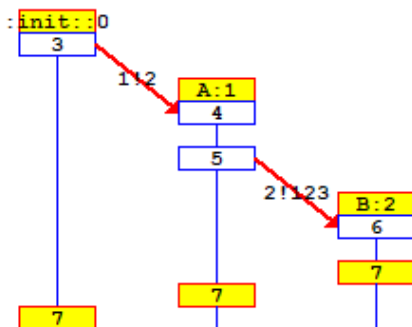
- Da li se radi o sinkronoj ili asinkronoj komunikaciji između procesa? Po čemu to zaključujete?
- Grafički prikažite rezultate simulacije (*Sequence Chart*). Koliko je procesa ukupno kreirano?
- Što je rezultat simulacije (*Simulation Output*)? U predviđene prostore upišite rješenja zadataka.

Rješenje:

a) Odgovor:

Veličina kanala je 1 što nam sugerira da je komunikacija asinkrona.

b) Grafički prikaz:



* Ukupno su kreirana tri procesa.

c) Odgovor:

x = 123

Zadatak 3:

Zadan je sljedeći Promela program koji opisuje jedan komunikacijski protokol:

```
/* Data Transfer Protocol */
mtype = { ini, ack, dreq, data, shutup, quiet, dead };
chan M = [1] of { mtype };
chan W = [1] of { mtype };
active proctype Mproc()
{
  W!ini;
  M?ack;
  timeout -> /* wait */
  if
  :: W!shutup
  :: W!dreq;
  M?data ->
  do
  :: W!data
  :: W!shutup;
  break
  od
  fi;
  M?shutup;
  W!quiet;
  M?dead
}
active proctype Wproc()
{
  W?ini;
  M!ack;
  do
  :: W?dreq ->
  M!data
  :: W?data ->
  skip
  :: W?shutup ->
  M!shutup;
  break
  od;
  W?quiet;
  M!dead
}
```

- Da li se radi o sinkronoj ili asinkronoj komunikaciji između procesa?
- Što se deklarira u prvoj liniji koda?
- Koliko je procesa deklarirano u programu? Navedite koji su to procesi, te što rade.
- Grafički prikažite rezultate simulacije (*Sequence Chart*).

U predviđene prostore upišite rješenja zadataka.

Rješenje:

a) Odgovor:

Komunikacija je asinkrona.

b) Odgovor:

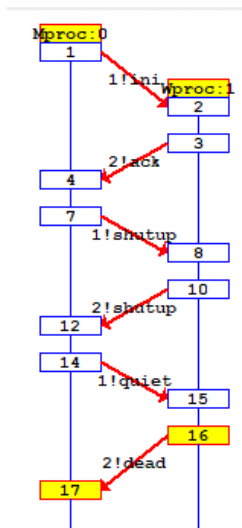
U prvoj liniji koda deklariraju se simboličke konstante.

c) Odgovor:

U programu se deklarirana dva procesa:

- a) Mproc – zadužen za započinjanje komunikacije
 - u slučaju da potvrda ne dođe, šalje shutup te dreg.
- b) Wproc – potvrđuje komunikaciju (ACK)

d) Grafički prikaz:



2. PETRIJEVE MREŽE

Petrijeve mreže predstavljaju grafički i matematički alat za modeliranje informacijskih sustava. Naziv su dobile prema njihovom izumitelju Adamsu Petriju. Primjenjuju se za opis i proučavanje različitih vrsta sustava – *asinkronih, konkurentnih, distribuiranih* i dr. Omogućuju definiranje i simuliranje različitih stanja i procesa u promatranim sustavima te opisivanje informacijskih tokova u sustavima.

Pri definiranju sustava uvjeta i događaja, pojam Petrijeve mreže uobičajeno se odnosi na mrežu mjesta i prijelaza. U takvoj mreži mjesta imaju značenje uvjeta, a prijelazi događaja. Struktura Petrijeve mreže opisuje se uređenom četvorkom $C = (P, T, I, O)$, pri čemu je:

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_n\} && \text{konačan skup mjesta, } n > 0, p_i \in P, \\ T &= \{t_1, t_2, \dots, t_m\} && \text{konačan skup prijelaza, } m > 0, t_j \in T, \\ P \cap T &= \emptyset \\ P \cup T &\neq \emptyset \\ I : T &\rightarrow P && \text{funkcija ulaza,} \\ O : T &\rightarrow P && \text{funkcija izlaza.} \end{aligned}$$

Funkcija ulaza i funkcija izlaza opisuju preslikavanje iz prijelaza u skupinu ulaza, odnosno izlaza. Pri tome vrijedi sljedeće:

- mjesto p_i je ulazno mjesto za prijelaz t_j ako je $p_i \in I(t_j)$, a izlazno mjesto ako je $p_i \in O(t_j)$;
- svako mjesto može biti višestruko ulazno, odnosno višestruko izlazno mjesto za neki prijelaz.

Broj pojavljivanja mjesta p_i kao ulaznog, odnosno izlaznog označuje se na sljedeći način:

$$\#(p_i, I(t_j)) \text{ i } \#(p_i, O(t_j)).$$

Funkcija ulaza i funkcija izlaza mogu se definirati kao preslikavanje mjesta u prijelaze na sljedeći način:

$$\begin{aligned} I : P &\rightarrow T \\ O : P &\rightarrow T, \end{aligned}$$

pri tome vrijedi:

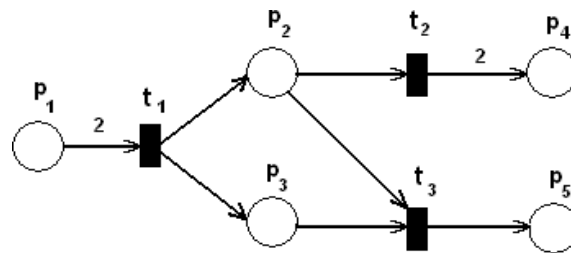
$$\begin{aligned} \#(t_j, I(p_i)) &= \#(p_i, O(t_j)) \\ \#(t_j, O(p_i)) &= \#(p_i, I(t_j)). \end{aligned}$$

Za prikazivanje strukture Petrijevih mreža primjenjuju se usmjereni grafovi. Oni se sastoje od niza čvorova (stanja i prijelaza) i usmjerenih grana (veze između čvorova). Primjer 1.

Grafički predočite strukturu Petrijeve mreže ako je zadano:

$$\begin{aligned} P &= \{p_1, p_2, p_3, p_4, p_5\} \\ T &= \{t_1, t_2, t_3\} \\ I(t_1) &= \{p_1\} && \#(p_1, I(t_1)) = 2 \\ I(t_2) &= \{p_2\} && \#(p_2, I(t_2)) = 1 \\ I(t_3) &= \{p_2, p_3\} && \#(p_2, I(t_3)) = 1 && \#(p_3, I(t_3)) = 1 \\ O(t_1) &= \{p_2, p_3\} && \#(p_2, O(t_1)) = 1 && \#(p_3, O(t_1)) = 1 \\ O(t_2) &= \{p_4\} && \#(p_4, O(t_2)) = 2 \\ O(t_3) &= \{p_5\} && \#(p_5, O(t_3)) = 1. \end{aligned}$$

Grafički prikaz:



Označavanje Petrijeve mreže

Graf Petrijeve mreže $G = (V, A)$ ekvivalentan je strukturi $C = (P, T, I, O)$ ako je zadovoljeno:

- » $V = \{v_1, v_2, \dots, v_s\}$ skup čvorova,
- » $A = \{a_1, a_2, \dots, a_r\}$ skup grana,
- » $V = P \cup T, P \cap T \neq \emptyset$,
- » $a_i = (v_j, v_k), v_j \in P \text{ i } v_k \in T \text{ ili } v_j \in T \text{ i } v_k \in P$.

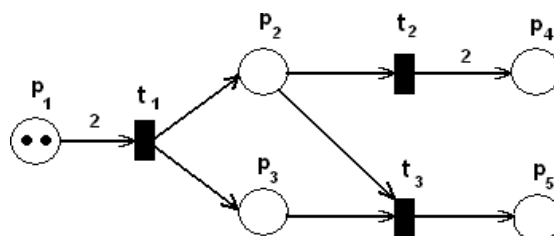
Označenu Petrijevu mrežu $M = (P, T, I, O, \mu)$ čini struktura mreže C s vektorom oznaka μ . Označavanje Petrijeve mreže provodi se pridruživanjem oznaka ' \bullet ' mjestima mreže, sa značenjem ispunjenosti uvjeta. Broj i raspored oznaka u mjestima može se izmijeniti nakon aktiviranja prijelaza u mreži.

» Označavanje Petrijeve mreže $C = (P, T, I, O)$ je funkcija $\mu : P \rightarrow N$ (gdje je N skup nenegativnih cijelih brojeva) koja se u vektorskom obliku predočuju ovako:

$$\mu = (\mu_1, \mu_2, \dots, \mu_i, \dots, \mu_n).$$

Pri tome μ_i odgovara broju oznaka u mjestu p_i , odnosno $\mu(p_i) = \mu_i$.

Ako je zadan sljedeći vektor oznaka $\mu = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) = (2, 0, 0, 0, 0)$ u *Primjeru 1*. raspored oznaka prikazan je na slici ispod.



» Dakle, svaka mreža je određena brojem i distribucijom oznaka.

» Mreža se izvodi realizacijom prijelaza, pri čemu ulazna mjesta gube oznake, a izlazna ih dobivaju.

Prijelaz $t_j \in T$ se u mreži $M = (P, T, I, O, \mu)$ može aktivirati ako je za svaki $p_i \in P$:

$$\mu(p_i) \geq \#(p_i, I(t_j)),$$

odnosno ako svako ulazno mjesto ima najmanje onoliko oznaka s koliko je grana povezano s prijelazom.

» Aktiviranje prijelaza generira novo stanje μ' tako da za svaki $p_i \in P$ vrijedi:

$$\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j)).$$

» U *Primjeru 1.* uz μ kao početno stanje može se izvesti samo prijelaz t_1 , jer je ispunjeno:

$$\begin{array}{lll} \mu(p_1) = 2 & \geq & \#(p_1, I(t_1)) = 2 \\ \mu(p_2) = 0 & \geq & \#(p_2, I(t_1)) = 0 \\ \mu(p_3) = 0 & \geq & \#(p_3, I(t_1)) = 0 \\ \mu(p_4) = 0 & \geq & \#(p_4, I(t_1)) = 0 \\ \mu(p_5) = 0 & \geq & \#(p_5, I(t_1)) = 0. \end{array}$$

Graf stanja i stablo dostupnosti

Primjer 2.

Za Petrijevu mrežu iz *Primjera 1.* i početno stanje $\mu^0 = (2, 0, 0, 0, 0)$ odredite skup dostupnih stanja, graf stanja, generirane sljedove stanja i prijelaza, te dostupnastanja.

Rješenje:

Skup dostupnih stanja $R(\mu)$ sadrži:

$$\mu^0 = (2, 0, 0, 0, 0)$$

$$\mu^1 = (0, 1, 1, 0, 0)$$

$$\mu^2 = (0, 0, 1, 2, 0)$$

$$\mu^3 = (0, 0, 0, 0, 1).$$

Generirani sljedovi stanja i prijelaza

$$su: (\mu_0, \mu_1, \mu_2) \quad (t_1, t_2)$$

$$(\mu_0, \mu_1, \mu_3) \quad (t_1, t_3).$$

Neposredno dostupna stanja su:

$$\mu^0 \text{ i } \mu^1$$

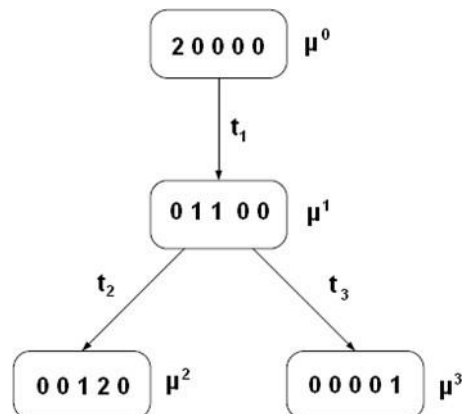
$$\mu^1 \text{ i } \mu^2$$

$$\mu^1 \text{ i } \mu^3.$$

Dostupna stanja su:

$$\mu^0 \text{ i } \mu^2$$

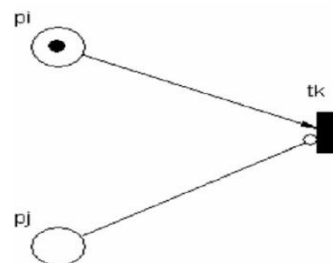
$$\mu^0 \text{ i } \mu^3.$$



Petrijeva mreža s inhibicijskom granom

Inhibicijska grana koja se nalazi između mjesta p_j i prijelaza t_k ima značenje negacije, pa se prijelaz t_k izvodi ako su sva “normalna” mjesta označena, a inhibicijsko mjesto nije označeno. Svaki prijelaz t_k može imati proizvoljan broj inhibicijskih mjesta. Inhibicijska grana označava se kružićem ('o') umjesto strelicom na simbolu prijelaza.

Primjena inhibicijske grane u modelima telekomunikacijskih procesa može značajno pojednostaviti strukturu modela, jer se često pojavljuju situacije u kojima se uz ispunjen uvjet provodi jedan, a uz neispunjen uvjet drugi skup akcija (*npr.* slobodan korisnik – zauzet korisnik). Na slici se nalazi prikaz inhibicijske grane između mjesta p_j i prijelaza t_k .



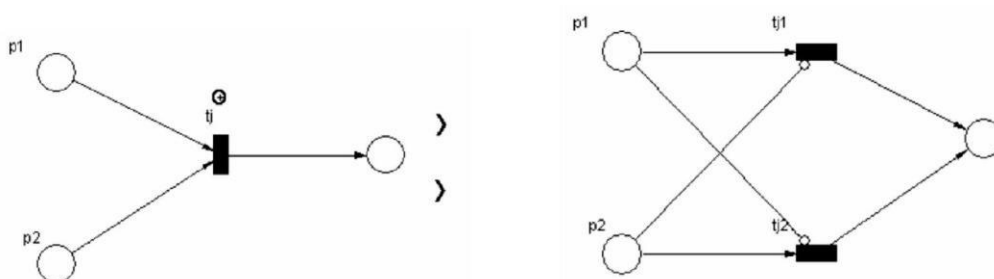
Slika 1. Inhibicijska grana

Petrijeva mreža s isključivim ILI prijelazom

Isključivi ILI prijelaz izvodi se ako je označeno samo jedno ulazno mjesto ili neparan broj ulaznih mjesta. Isključivi ILI prijelaz s k ulaznih mjesta može se opisati s k prijelaza, od kojih je svaki prijelaz povezan s mjestima na sljedeći način:

- svako ulazno mjesto povezano je s jednim od prijelaza “normalnom” granom, a s ostalim inhibicijskom granom;
- izlazno mjesto za sve prijelaze je zajedničko i odgovara izvornom izlaznom mjestu isključivog ILI prijelaza.

Primjer pretvorbe ILI prijelaza sa dva ulazna mjesta prikazan je Slikom 2.



Slika 2. Pretvorba ILI prijelaza

Taj primjer pokazuje da uporaba složenijih prijelaza pojednostavljuje strukturu mreže. Tipična je primjena takvog prijelaza u situacijama u kojima se modelira radnja za izvedbu koje je dovoljna ispunjenost samo jednog od preduvjeta.

Obilježja Petrijevih mreža

a) **Aktivnost:** Svojstvo aktivnosti (životnosti) odnosi se na mogućnost izvedbe prijelaza. Aktivna mreža isključuje mogućnosti blokiranja ili potpunog zastoja u modeliranom sustavu. U aktivnim mrežama ne postoje prijelazi koji se nikada ne izvode ili stanja u kojima se ne može izvesti niti jedan prijelaz.

b) **Sigurnost:** Mjesto je sigurno ako broj žetona u bilo kojoj oznaci skupa $R(\mu^0)$ ne prelazi broj jedan. Petrijeva mreža je sigurna ako je svako mjesto sigurno.

Ograničenost: Koncept ograničenosti se može gledati kao pojednostavljena sigurnost. Mjesto je ograničeno s ograničenjem k , ako broj žetona ne prelazi k ni u kojoj oznaci $R(\mu^0)$. Petrijeva mreža je k -ograničena ako je svako mjesto k -ograničeno.

c) **Konzervacija oznaka:** Petrijeva mreža je strogo konzervacijska ukoliko je ukupan broj oznaka u svakom stanju skupa $R(\mu^0)$ konstantan.

d) **Reverzibilnost:** Petrijeva mreža je reverzibilna ako se iz svakog stanja μ' može vratiti u početno stanje μ , odnosno ako je početno stanje dostupno iz svakog stanja.

e) **Perzistentnost:** Kaže se da je Petrijeva mreža perzistentna ako za bilo koja dva prijelaza, aktiviranje jednog prijelaza ne onemogućuje drugi prijelaz. U perzistentnoj mreži, nakon što se određeni prijelaz omogući, on ostaje omogućen sve dok se ne aktivira.

Preporučena literatura:

- prezentacije i bilješke s auditornih vježbi i predavanja
- Lovrek, I., "Modeli telekomunikacijskih procesa: teorija i primjena Petrijevih mreža", Zagreb, Školska knjiga, 1997.

Priprema za vježbu:

Prije vježbi potrebno je proučiti predložak za vježbu te preporučenu literaturu.

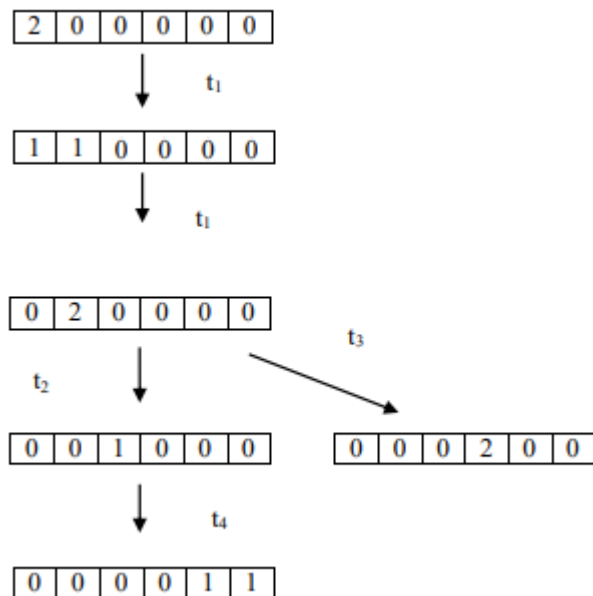
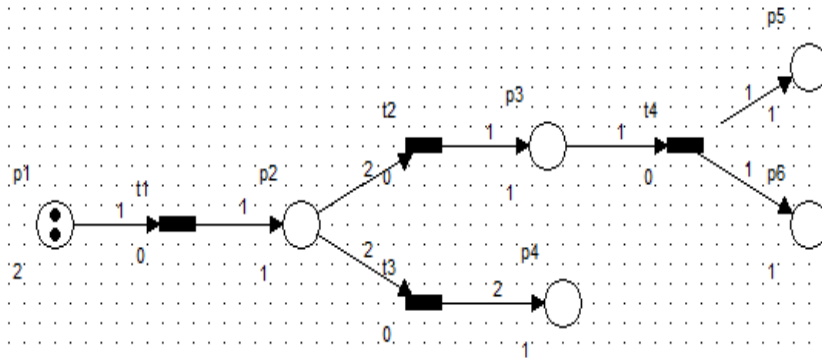
Cilj vježbe:

Cilj ove vježbe je upoznavanje s primjenom Petrijevih mreža pri modeliranju komunikacijskih protokola.

ZADACI

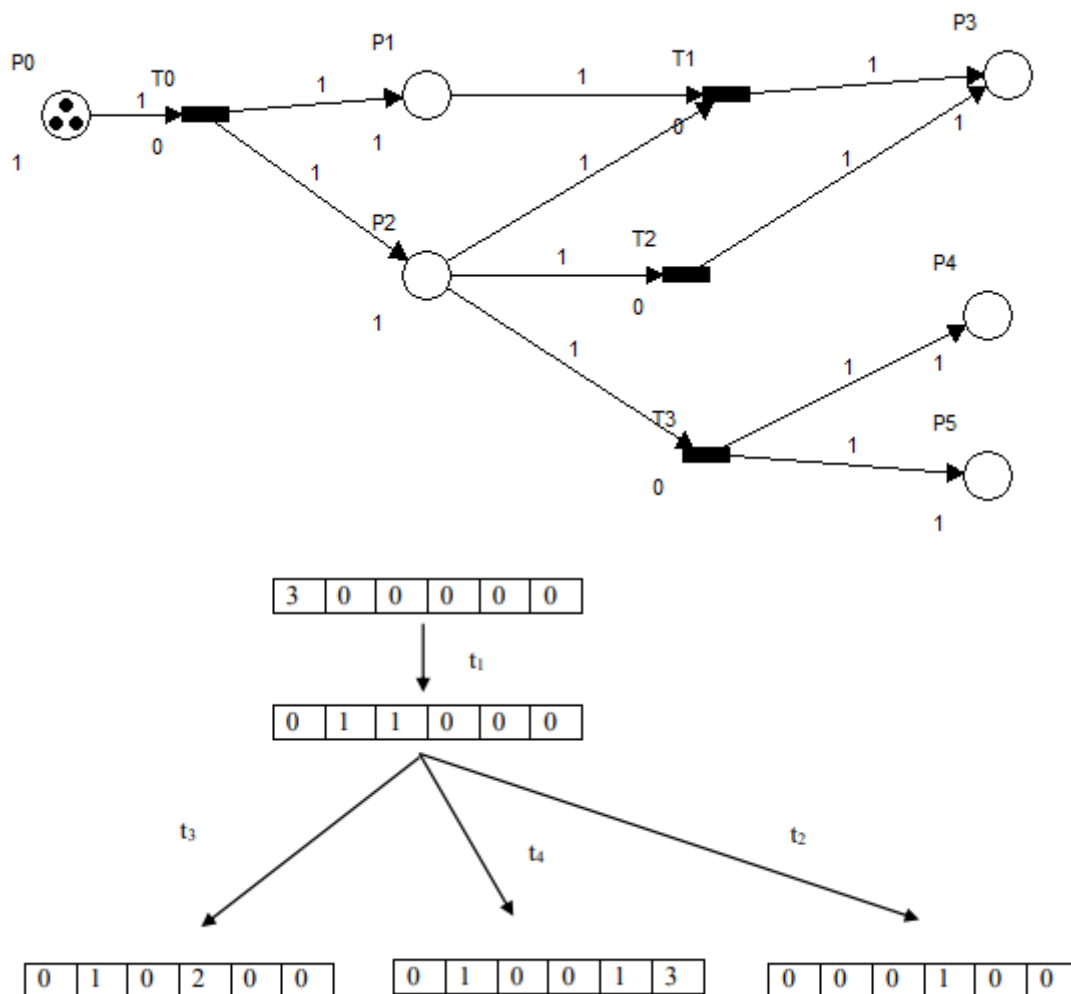
Zadatak 4:

Rješenje:



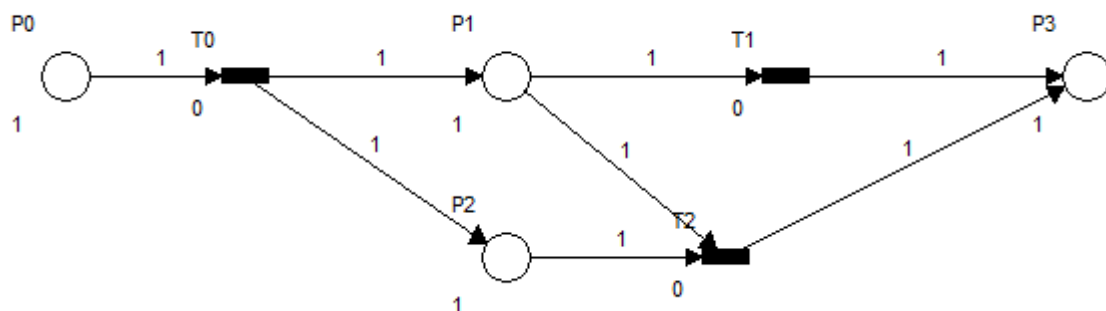
Zadatak 5:

Rješenje:



Zadatak 6:

Rješenje:



3	0	0	0
---	---	---	---

t_1



0	1	1	0
---	---	---	---

t_2



0	0	1	1
---	---	---	---

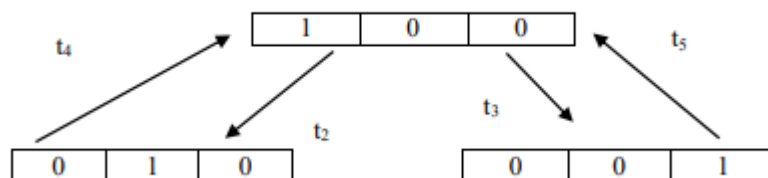
t_3



0	0	0	1
---	---	---	---

Zadatak 7:

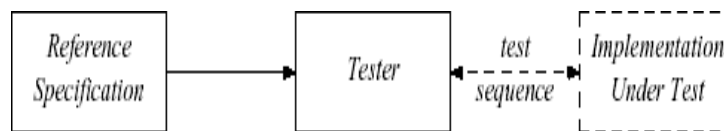
Rješenje:



- a) Ograničenost - Koncept ograničenosti se može gledati kao pojednostavljena sigurnost. Mjesto je ograničeno sa ograničenjem k , ako broj oznaka (žetona) ne prelazi k u bilo kojoj oznaci $R(\mu_0)$. Petrijeva mreža je k -ograničena ako je svako mjesto k -ograničeno. Ova mreža je ograničena zato što broj oznaka u svakom mjestu za svako stanje 1.
- b) Aktivnost - Svojstvo životnosti (aktivnosti) se odnosi na mogućnost izvedbe prijelaza. Aktivna mreža isključuje mogućnost blokiranja ili potpunog zastoja u modeliranom sustavu koji se manifestira postojanjem prijelaza koji se nikad ne izvodi ili stanja u kojemu se ne može izvesti nijedan prijelaz. Ova mreža nije aktivna zato što se u niti jednom stanju prijelaz t_1 ne može izvesti.
- c) Reverzibilnost - Petrijeva mreža je reverzibilna ako se iz svakog stanja μ' može vratiti u početno stanje μ , odnosno ako je početno stanje dostupno iz svakog stanja. Vidimo da je ova mreža reverzibilna zato što se iz oba stanja mreža može vratiti u početno stanje.

3. FORMALNE METODE SPECIFIKACIJE I SIMULACIJSKO MODELIRANJE PROTOKOLA

Razvoj raspodijeljenih računalnih sustava i njihovo oblikovanje je složen zadatak. Zbog sve većih zahtjeva na funkcionalnost sustava raste njihova kompleksnost. Povećanjem kompleksnosti povećava se i broj pogrešaka do kojih može doći u procesu njihovog razvoja. Stoga je za osiguranje ispravnog funkcioniranja računalnih sustava nužno detaljno provjeravanje, odnosno testiranje rada tih sustava.



Slika 1. Testiranje sustava

Svrha dizajna pri razvoju nekog sustava bit će postignuta ako postoji mogućnost testiranja izvedbe sustava kako bi se odredilo da li izvedba odgovara danim zahtjevima i standardima (primjena standardizirane test sekvence).

Pri razvoju protokola razlikujemo:

- neformalne
- poluformalne i
- formalne metode razvoja protokola.

Formalne metode koriste se za razvoj nedvosmislenih, cjelovitih i točnih specifikacija protokola te stvaraju osnovu za daljnju verifikaciju protokola. Glavna prednost ovih metoda u odnosu na neformalne i poluformalne je precizna rezultirajuća specifikacija protokola kod koje se može detaljno analizirati kompletnost i konzistentnost modela.

Formalne metode za specifikaciju protokola se temelje na tri tipa modela:

1. modeli prijelaza stanja – konačni automati, Petrijeve mreže i njene varijante
2. modeli programskih jezika
3. hibridni modeli.

Ovisno o namjeni i svojstvima modeliranih sustava u primjeni postoji veliki broj različitih vrsta modela. Simulacijski modeli, modeli dinamičkih sustava, su modeli čije se stanje mijenja tijekom vremena. Oni omogućavaju ispravan prikaz i efikasno izvođenje pomaka vremena. Također, omogućuju paralelizam pri odvijanju aktivnosti, te opis konkurentskih procesa.

Nad komponentama simulacijskog modela vrši se analiza strukture i načina rada sustava, modeliranje, programiranje te simulacijski proces.

Prvi korak simulacijskog modeliranja je izgradnja konceptualnih simulacijskih modela. Konceptualni modeli stvaraju se na temelju strukture i logike rada sustava koji se modelira, a prikazuju se simbolima točno definiranog značenja.

Konceptualni modeli opisuju elemente određenog sustava i njihovo međudjelovanje, te omogućuju strukturiranje problema. Sadrže grubi opis sustava. Metode za konceptualno simulacijsko

modeliranje dijele se na:

- **grafičke metode**
 - prikaz skupa simbola povezanih u dijagrame (prednosti: konceptualno blisko povezani elementi prikazuju se fizički blizu, omogućuju vizualizaciju modeliranog sustava) - primjer grafičkih metoda: Petrijeve mreže;
- **proceduralne metode**
 - vrsta pseudokoda - za sekvencijalni opis procedura koje opisuju rad sustava;
- **kombinirane**
 - kombinacija grafičkih metoda i proceduralnih metoda koje detaljno opisuju grafičke module;
- **metode interaktivnih upitnika**
 - softverski sustavi s osnovnim znanjima o simulaciji.

Jedna od metoda konceptualnog simulacijskog modeliranja su i Petrijeve mreže. To je grafička metoda modeliranja koja omogućuje izgradnju konceptualnog modela određenog sustava čije se ponašanje želi simulirati.

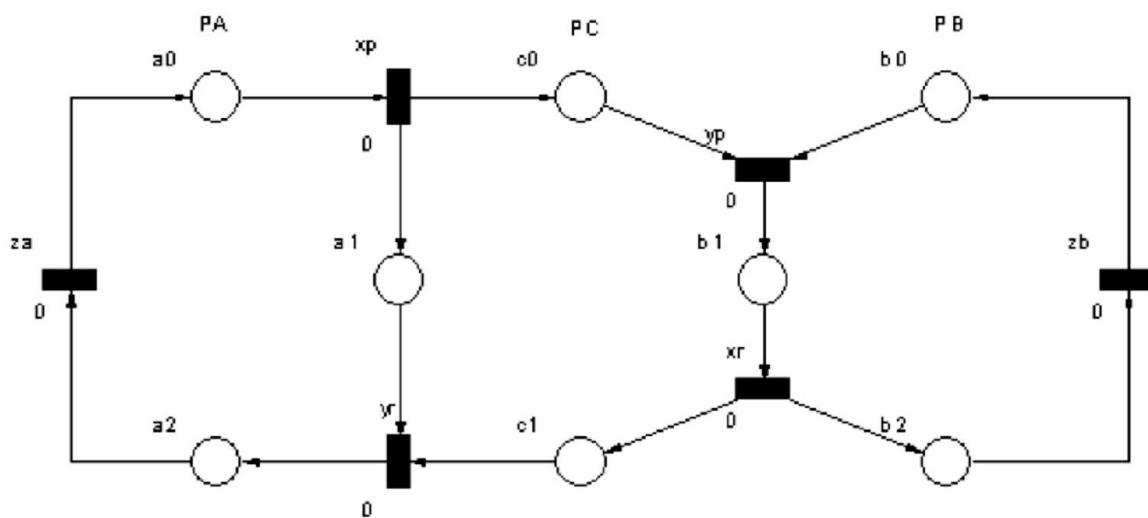
Petrijeve mreže posjeduju bitne karakteristike za konceptualno modeliranje - jednostavnost, nedvosmislenost, mogućnost verifikacije općih svojstva protokola i vizualizaciju modeliranog sustava. Ipak, one nisu uvijek prikladne za opis kompleksnih protokola budući da mogu sadržavati previše stanja i prijelaza. Tada se za modeliranje koriste druge formalne metode specifikacije protokola - modeli programskih jezika ili hibridni modeli.

Model osnovnoga komunikacijskog protokola

Svaki se proces modelira Petrijevom mrežom. Procesi se međusobno povezuju zajedničkim mjestima tako da se kanal također opisuje Petrijevom mrežom. Zajedničko mjesto označava postuvjet za prijelaz koji odašilje informacijsku jedinicu, a preduvjet za prijelaz kojim se ta informacijska jedinica prima.

U modelu osnovnog komunikacijskog protokola dva procesa međusobno izmjenjuju poruke i potvrde. Svakom procesu (A i B) i kanalu (C) pridjeljuje se odgovarajuća Petrijeva mreža (P_A, P_B i P_C) s mjestima koja opisuju uvjete i prijelaze (Slika 3.).

Sustav komunicirajućih procesa opisan je jednom mrežom koja je nastala povezivanjem pojedinačnih Petrijevih mreža. Komunikacijski kanal izravno povezuje komunicirajuće procese. Pri tom su utvrđeni uvjeti koji opisuju kanal.



Slika 2. Izvorna Petrijeva mreža za model protokola

Mjesta (opisuju uvjete):	Prijelazi (opisuju događaje):
a ₀ - pripravnost za predaju poruke	x _p – predaja poruke
a ₁ – čekanje potvrde	y _r – prijam potvrde
a ₂ – primljena potvrda	z _a – unutrašnji prijelaz
b ₀ – pripravnost za prijam poruke	y _p – prijam poruke
b ₁ – primljena poruka	x _r – predaja potvrde
b ₂ – predana poruka	z _b – unutrašnji prijelaz
c ₀ – poruka na kanalu	
c ₁ – potvrda na kanalu	

Struktura Petrijeve mreže opisuje ovisnost uvjeta i događaja i njihov međusobni utjecaj. To se može nazvati statičkim modelom komunikacijskog protokola. Dinamička komponenta uvodi se izvedbom Petrijeve mreže uz zadane početne uvjete.

Za naš primjer sa Slike 3. to je označavanje:

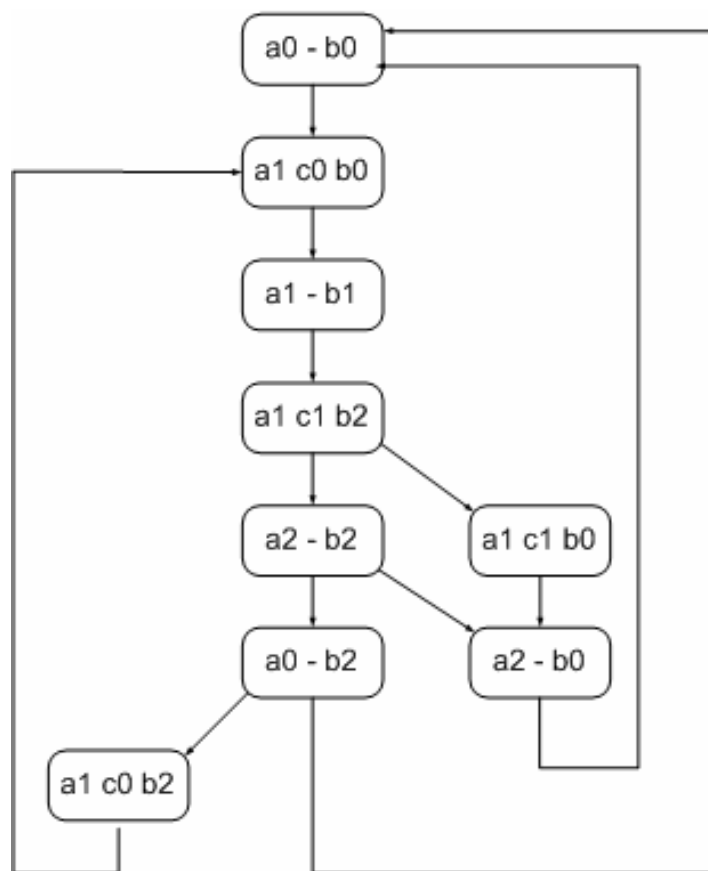
$$\mu_0 = (\mu_{a0}, \mu_{a1}, \mu_{a2}, \mu_{b0}, \mu_{b1}, \mu_{b2}, \mu_{c0}, \mu_{c1}) = (1, 0, 0, 1, 0, 0, 0, 0), \text{ odnosno}$$

ispunjeni uvjeti a₀ i b₀.

U tom stanju može se izvesti samo prijelaz x_p. Ulazno mjesto gubi jednu oznaku (a₀), a izlazna mjesta dobivaju po jednu oznaku (a₁, c₀), te se uspostavlja novo stanje $\mu_1 = (0, 1, 0, 1, 0, 0, 1, 0)$, u kojemu su označena mjesta a₁, b₀ i c₀.

U strukturnom smislu mreža je restriksijska, jer nema višestrukog povezivanja mjesta i

prijelaza, ni vlastitih petlji. Svako mjesto ima samo jedan ulazni i samo jedan izlazni prijelaz što odgovara pojmu označenoga grafa.



Slika 3. Graf stanja izvorne Petrijeve mreže za model protokola

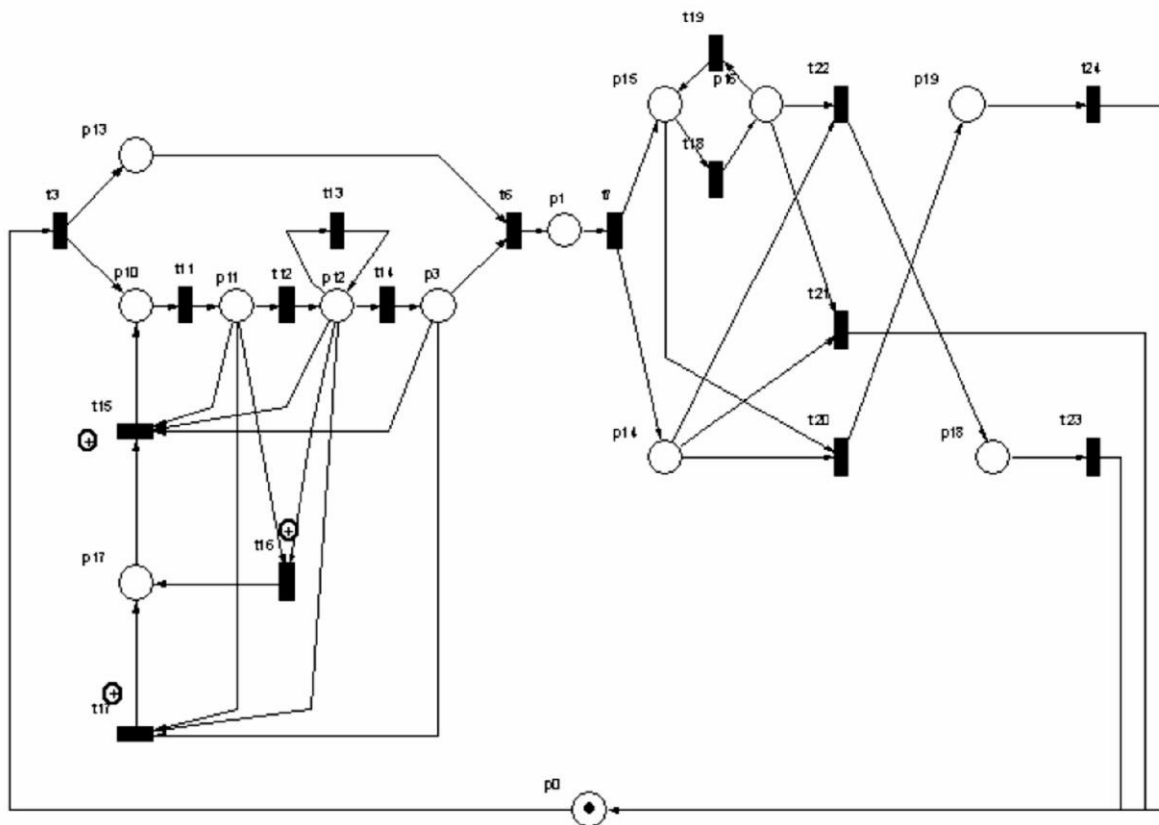
Modeli poziva i usluga

Polazeći od elementarne Petrijeve mreže može se provesti sustavan postupak razvoja algoritama obrade poziva. Svaki složeniji izvedeni model zadržava dobra svojstva sigurnosti i aktivnosti mreže. Na Slici 4. nalazi se formalna specifikacija poziva Petrijevom mrežom.

U ovom primjeru riješena su dva pitanja - odustajanje od poziva ili nekorektno biranje tijekom uspostavljanja komunikacije i prekidanje komunikacije na način uobičajen za govornu komunikaciju u javnoj telekomunikacijskoj mreži.

Razlikuje se postupak prekida veze pokrenut od A ili B korisnika. Prekid koji pokreće korisnik A je konačan. Ako korisnik B prekida komunikaciju, on je može nastaviti ponovnim podizanjem mikrotelefonske komunikacije (slušalice) u određenom vremenskom roku, odnosno, korisnik B može odustati od prekida komunikacije.

Tipične oznake koje se upotrebljavaju pri specifikaciji govorne komunikacije su oznake za iniciranje i prihvatanje poziva (podizanje mikrotelefonske kombinacije) i za prekid (polaganje mikrotelefonske kombinacije).



Slika 4. Formalna specifikacija poziva Petrijevom mrežom

Oznake:

t3	započinjanje poziva	p0	korisnici pripravi za poziv
t6	javljanje B (B prihvća poziv)	p1	komunikacija (izmjena informac. između korisnika)
t7	A pokreće prekid	p3	A pozvao B
t11	A pokreće komunikaciju	p10	A mirno stanje
t12	A bira prvu znamenku B broja t13	p11	A može slobodno birati
t14	A bira ostale znamenke B broja	p12	A u stanju biranja
t14	A bira zadnju znamenku B broja	p13	B mirno stanje.
t15	A prekida prije ostvarene komunikacije s B	p14	A u ostvarenoj komunikaciji s B
t16	A bira pogrešnu znamenku (nepostojeći B broj)	p15	B u ostvarenoj komunikaciji s A
t17	A isteklo vrijeme za biranje znamenke	p16	B prekinuo komunikaciju
t18	B prekida za vrijeme komunikacije s A	p17	A blokiran
t19	B odustaje od prekida	p18	A blokiran nakon komunikacije s B
t20	A prekida za vrijeme komunikacije s B	p19	B blokiran nakon komunikacije s A.
t21	A prekida nakon što je B prekinuo		
t22	B isteklo vrijeme za odustajanja od prekida		
t23	A prekida za vrijeme blokade		
t24	B prekida za vrijeme blokade		

Preporučena literatura:

- prezentacije i bilješke s auditornih vježbi i predavanja
- Lovrek, I., "Modeli telekomunikacijskih procesa: teorija i primjena Petrijevih mreža", Zagreb, Školska knjiga, 1997.

Priprema za vježbu:

Prije vježbi potrebno je proučiti predložak za vježbu te preporučenu literaturu.

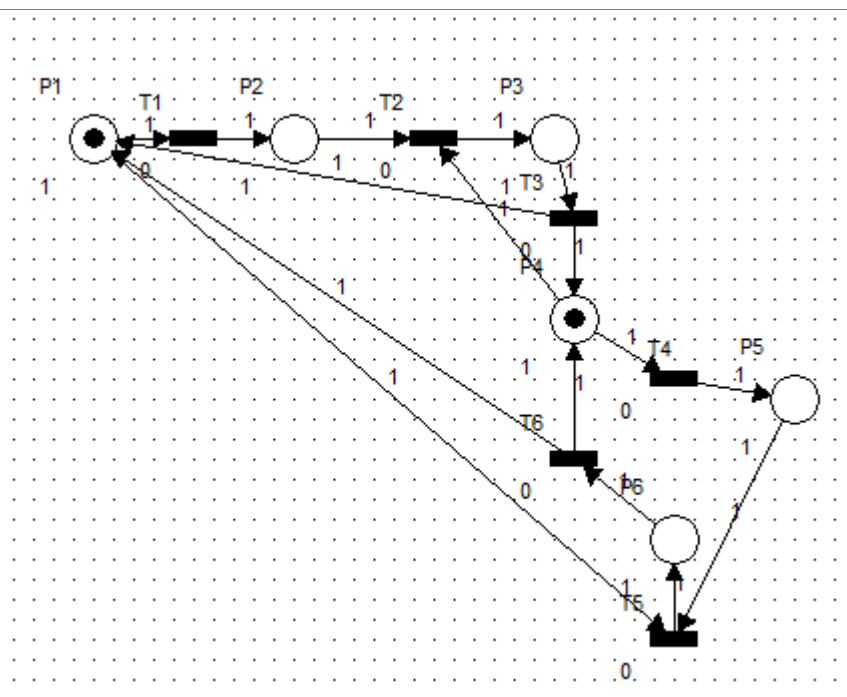
Cilj vježbe:

Cilj ove vježbe je upoznavanje s primjenom Petrijevih mreža pri modeliranju komunikacijskih protokola te provjera tako stvorenih modela u SPIN-u.

ZADACI

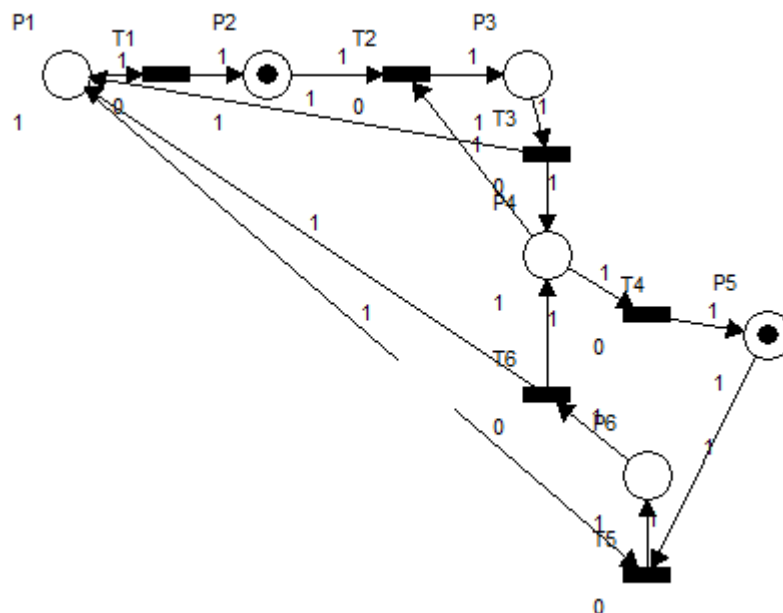
Zadatak 8:

Rješenje:



Step: 2 / Time: 0 =====>
Error: Deadlock at Time: 0 ms, Step: 2

Novi raspored oznaka u mreži:



Zadatak 9:

Rješenje:

```
#define Place 255

Place p1, p2, p3;
Place p4, p5, p6;
#define inp1(x)      (x>0) -> x=x-1
#define inp2 (x,y)   (x>0&&y>0) -> x = x-1; y=y-1
#define out1(x)      x=x+1
#define out2(x,y)    (); y=y+1
init
{ p1=1;p4=1; /* inicijalizacija */
do
/*t1*/ :: atomic { inp1(p1) -> out1(p2) }
/*t2*/ :: atomic { inp2(p2,p4)-> out1(p3) }
/*t3*/ :: atomic { inp1(p3) -> out2(p1,p4) }
/*t4*/ :: atomic { inp1(p4) -> out1(p5) }
/*t5*/ :: atomic { inp2(p1,p5)-> out1(p6) }
/*t6*/ :: atomic { inp1(p6) -> out2(p1,p4) }
od
}
```


Zadatak 10:**Rješenje:**

$p_1 = 0, p_2 = 1, p_4 = 0, p_5 = 1$ te $p_6 = 0$

Ime i prezime studenta:	Broj indeksa:	Datum:	Pregledano:
Damir Bašić		27.1.2023.	