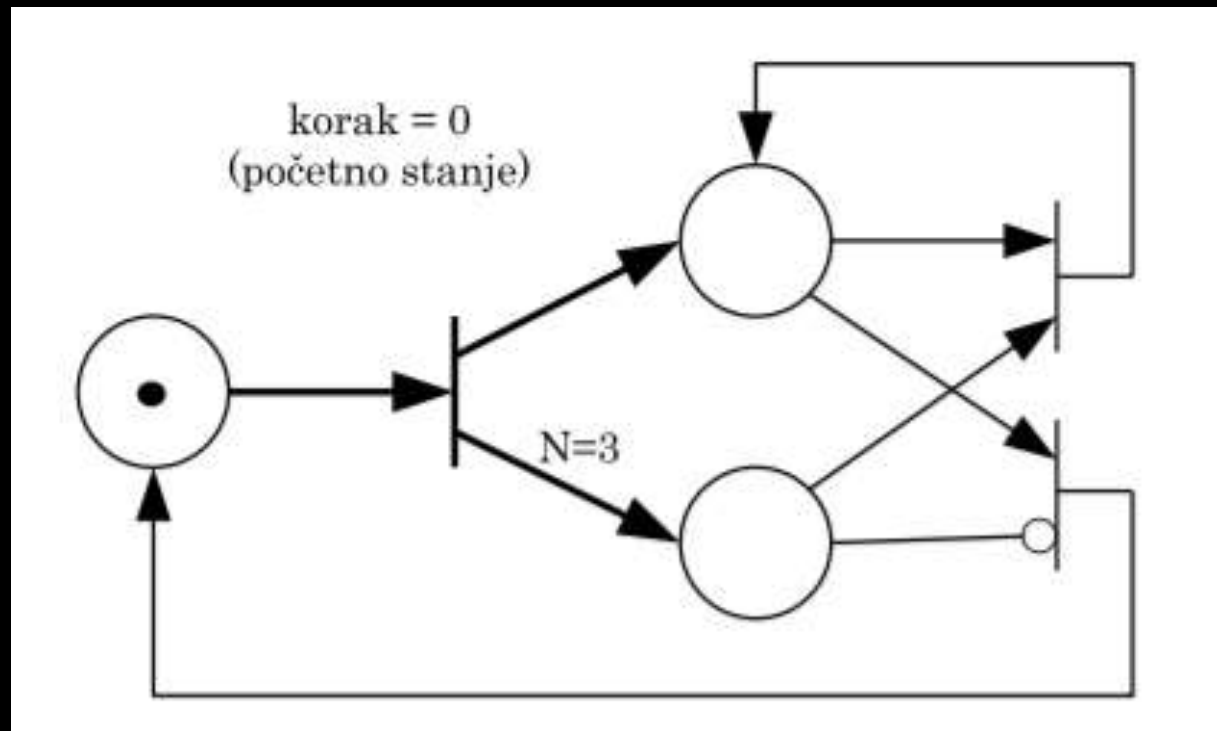


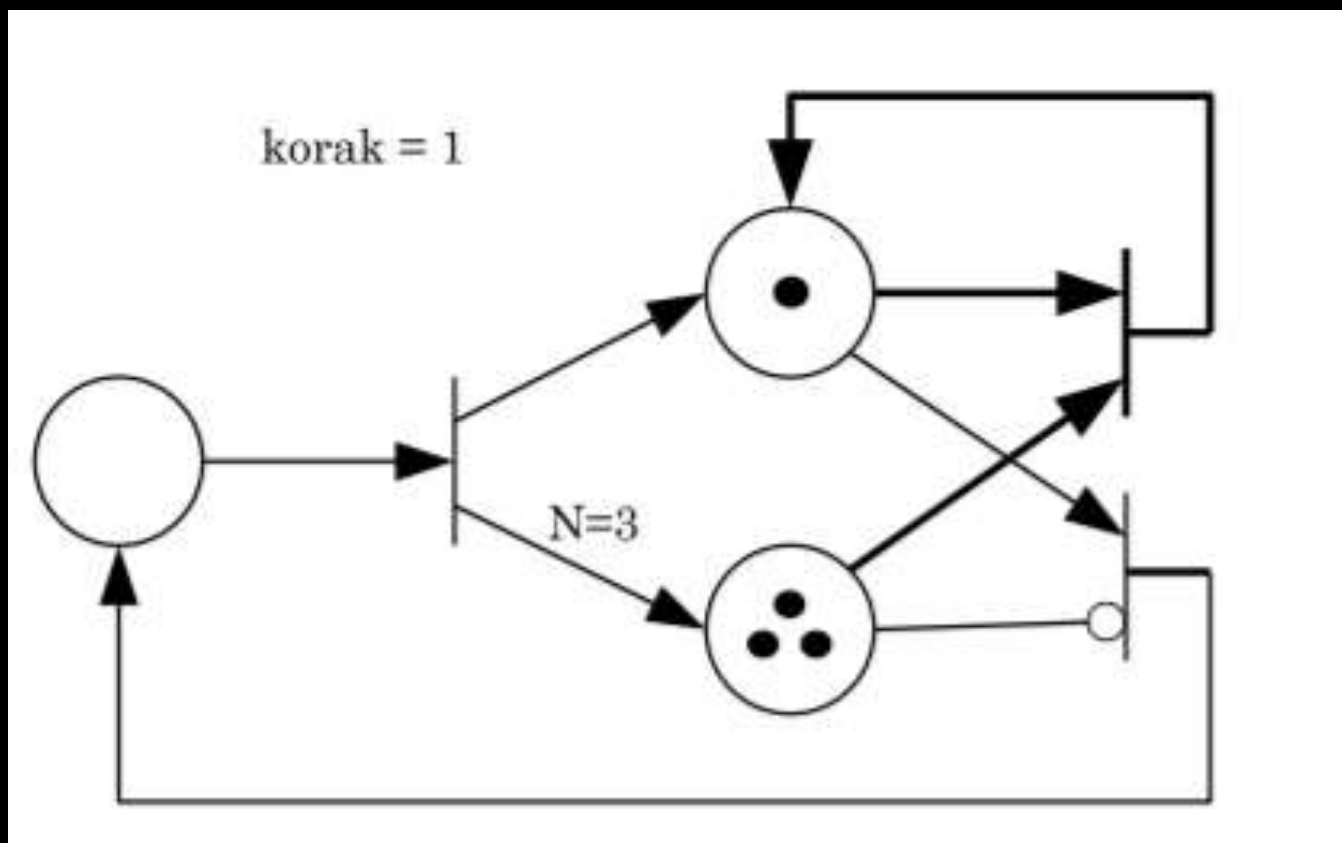
Petrijeve mreže

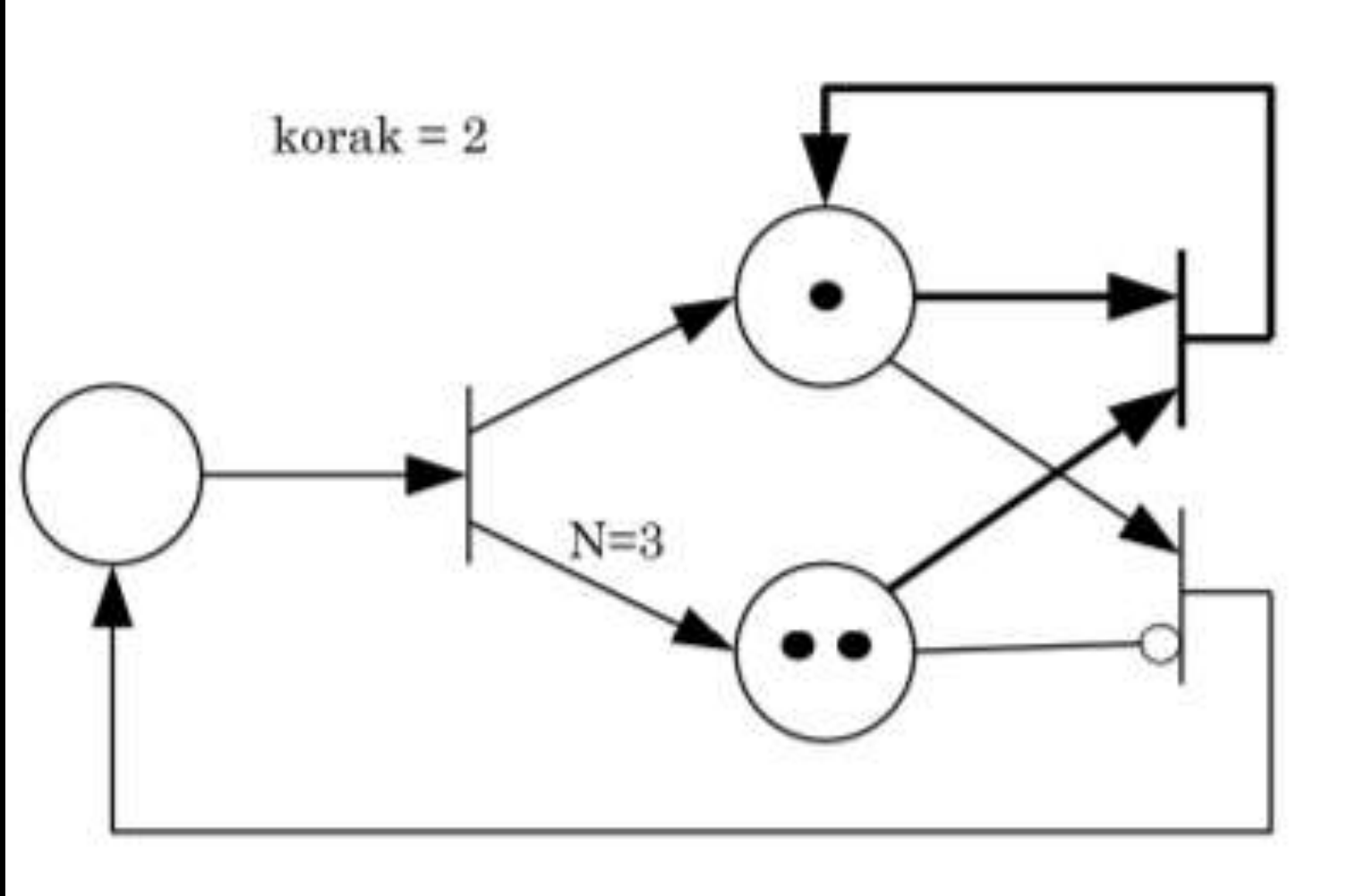
i dodatno u zadacima ima i uml
dijagrama

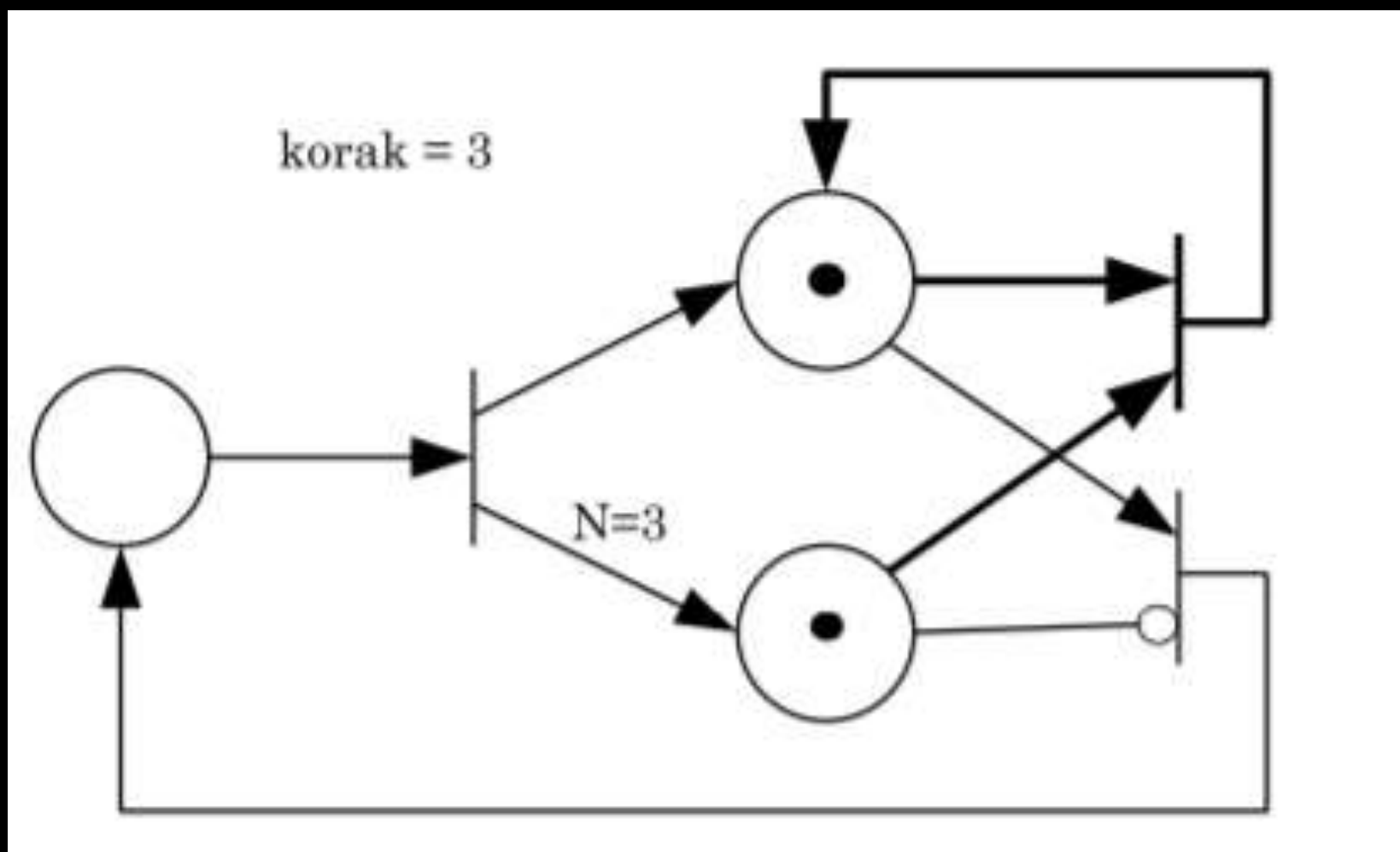
Kako će se ova mreža ponašati u koracima 2..5?

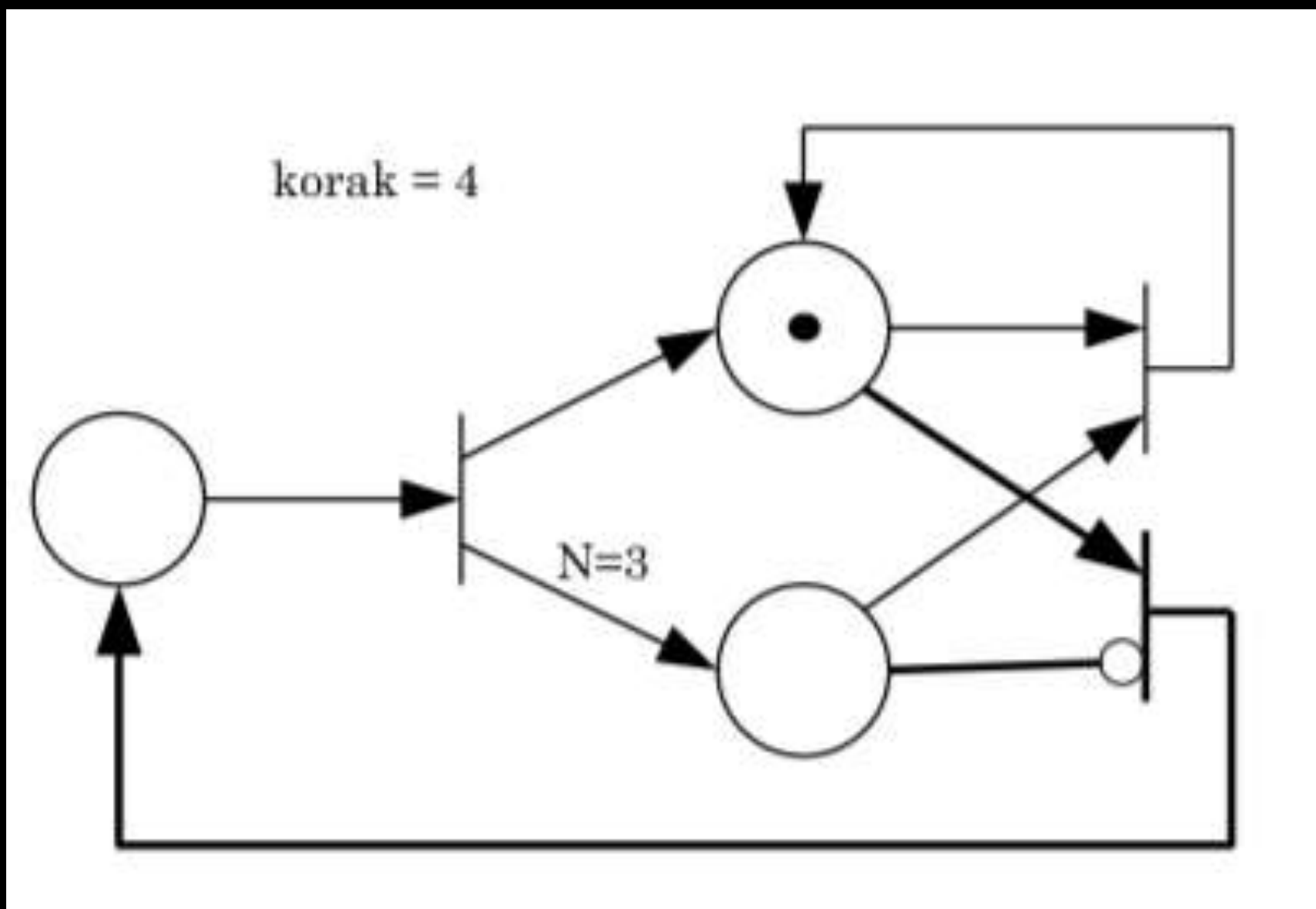


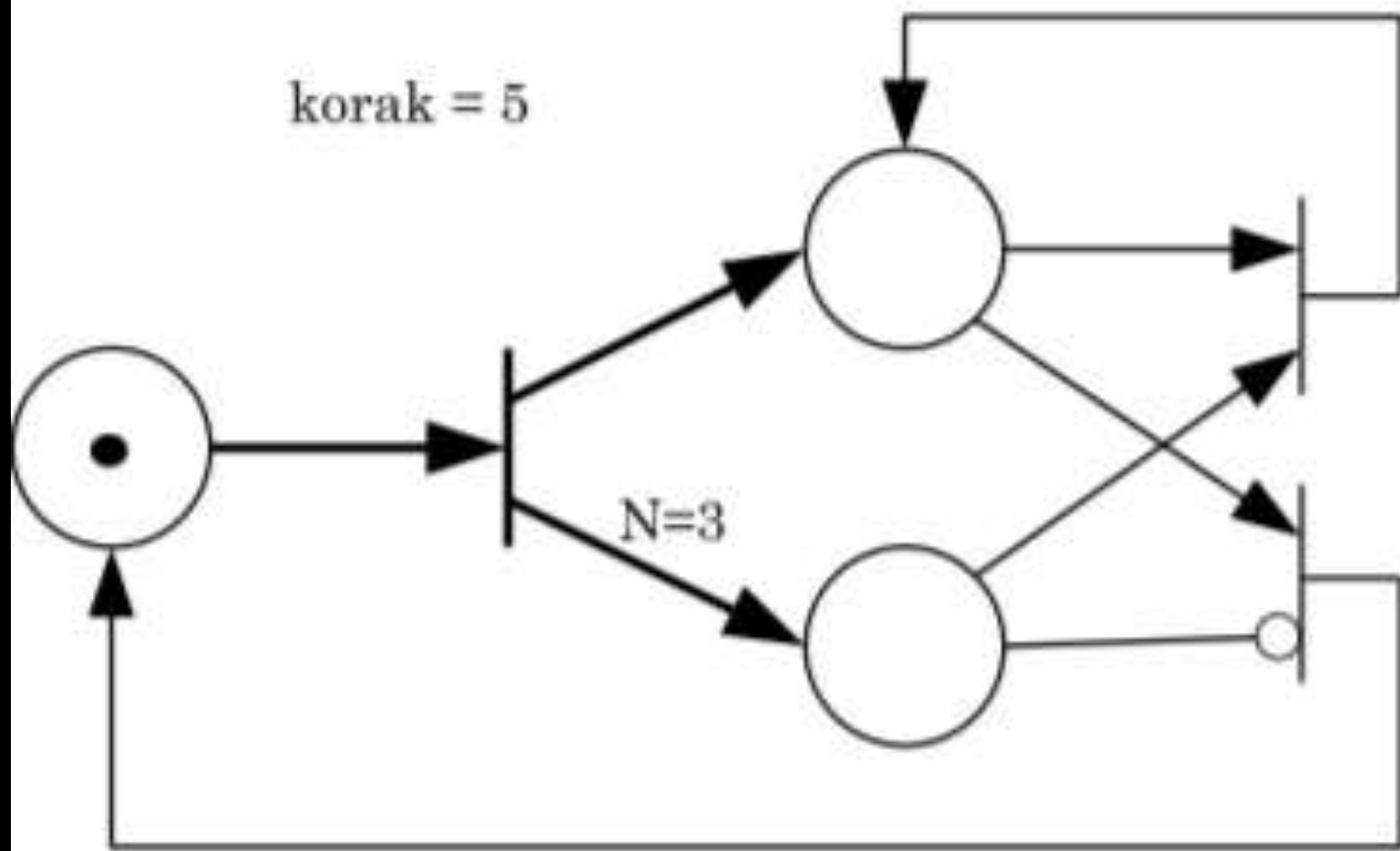
svaki korak će bit na jednom slideu











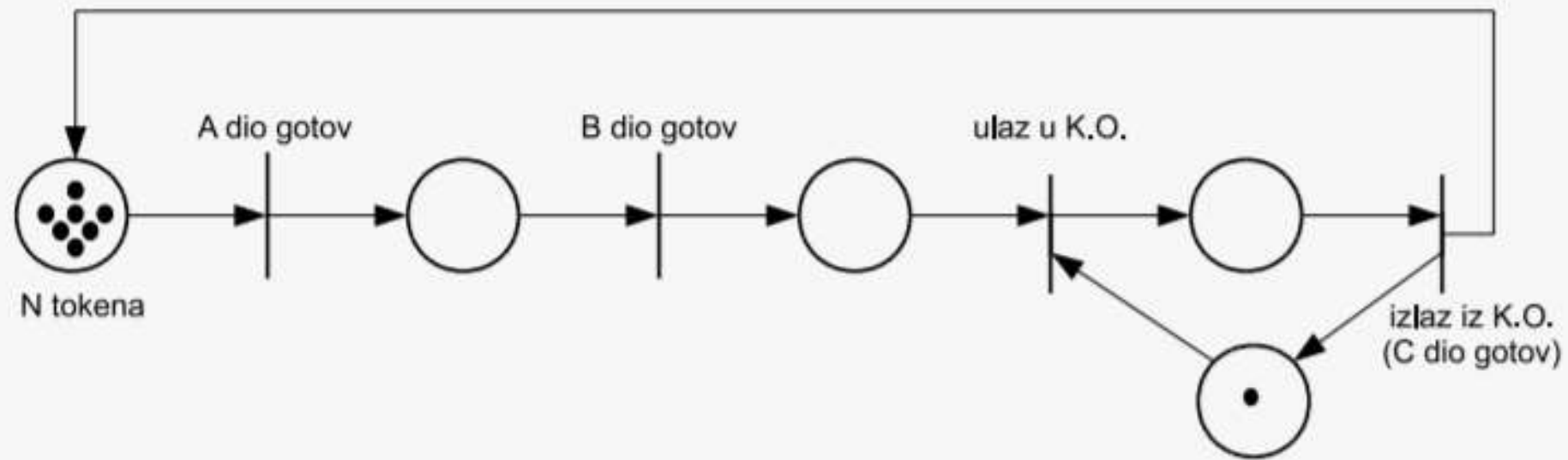
Primjer 2.2. Primjeri s Petrijevim mrežama

Skupina poslova

Za skup od N dretvi poznato je da obavljaju isti posao koji se sastoji od dijelova A, B i C (svaka dretva radi A dio, pa B dio, pa C dio te ponovno ispočetka A, B, C, ...). Prva dva dijela, A i B, razne dretve mogu obavljati paralelno, ali dio C moraju obaviti međusobno isključivo (u kritičnom odsječku). Modelirati sustav Petrijevom mrežom, gdje svaka značka

u početnom mjestu označava po jednu dretvu.

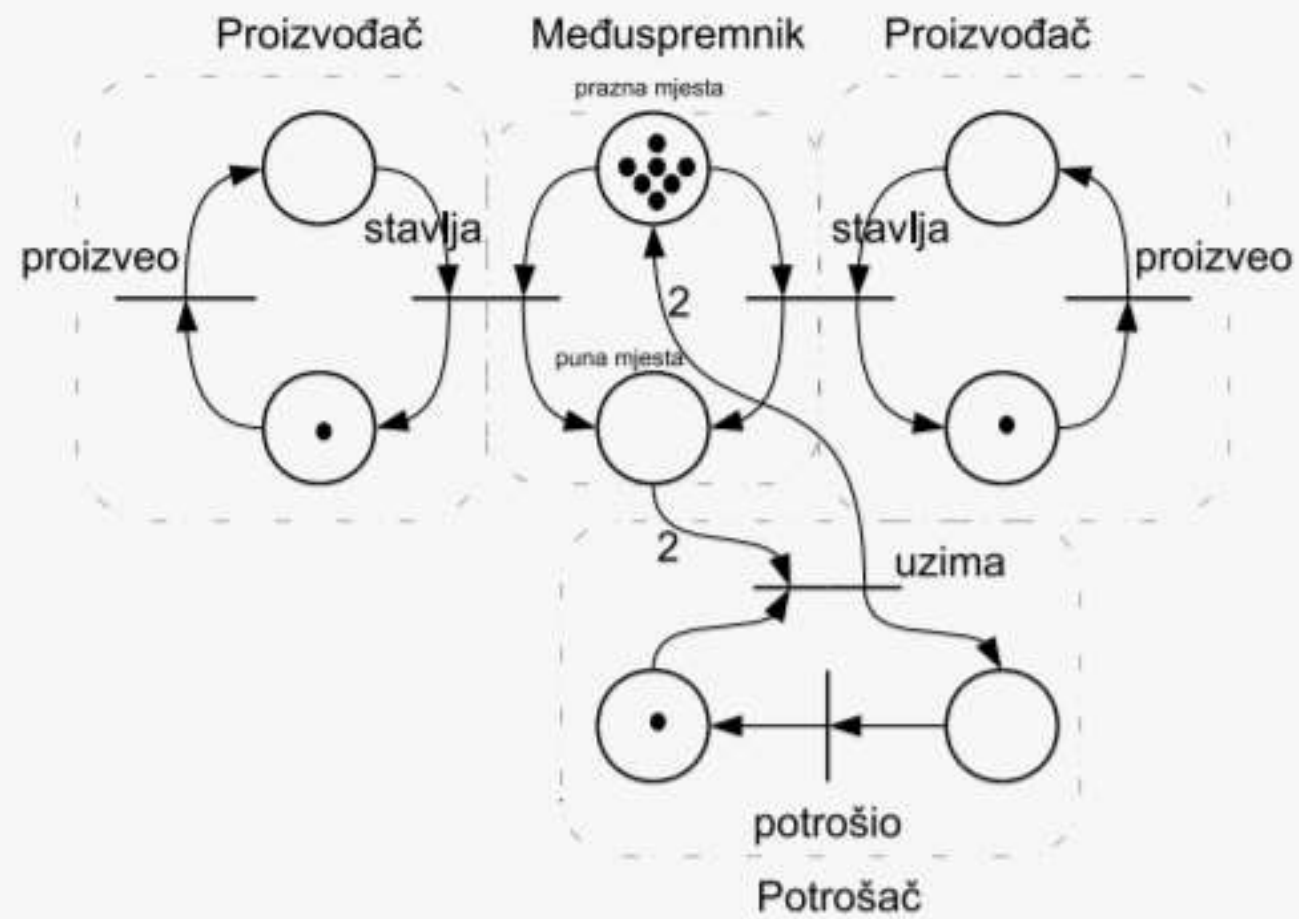
Rješenje:



Slika 2.10.

Proizvođači i potrošač

U sustavu se nalaze ciklički zadaci dva proizvođača i jednog potrošača. Proizvođači stvaraju poruke i stavljaju ih u prazna mjesta međuspremnik (ako je on pun, onda čekaju da se oslobodi jedno mjesto). Potrošač uzima dvije poruke iz međuspremnik te ih procesira (ako nema bar dvije poruke u međuspremniku onda čeka da se poruke pojave). Modelirati sustav Petrijevom mrežom. Veličina međuspremnik je 7 poruka.



Slika 2.11.

Kritični odsječak

Dva ciklička zadatka P_1 i P_2 u svojem izvođenju prolaze kroz kritični odsječak u kojem koriste zajedničke podatke te potom nastavljaju u nekritičnome. Ulazak u kritični odsječak je zaštićen odgovarajućim sinkronizacijskim mehanizmom tako da u njega uvijek može ući samo jedan zadatak. Trajanje kritičnog odsječka za P_1 je u granicama od 5 do 10 jedinica vremena, a za P_2 u granicama od 10 do 15. Trajanje nekritičnog odsječka za P_1 je u granicama od 5 do 25 jedinica vremena, a za P_2 od 20 do 40. Za zadani sustav izgraditi (grafički prikazati) vremensku Petrijevu mrežu te napisati tablicu s opisom prijelaza.

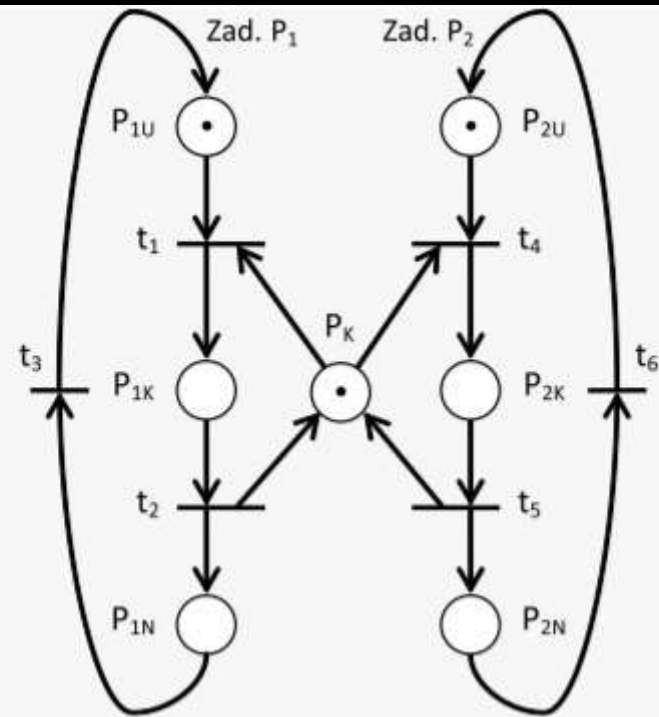
Rješenje:

Stanja su označena s P_{xU} , P_{xK} te P_{xN} sa značenjima:

P_{xU} – zadatak x čeka na ulaz u kritični odsječak

P_{xK} – zadatak x u kritičnom odsječku

P_{xN} – zadatak x u nekritičnom odsječku



Slika 2.13.

Tablica 2.2. table

prijelaz	ulazna mjesta	vremenska ograničenja
t_1	P_{1U}, P_K	–
t_2	P_{1K}	$[time(P_{1K}) + 5, time(P_{1K}) + 10]$
t_3	P_{1N}	$[time(P_{1N}) + 5, time(P_{1N}) + 25]$
t_4	P_{2U}, P_K	–
t_5	P_{2K}	$[time(P_{2K}) + 10, time(P_{2K}) + 15]$
t_6	P_{2N}	$[time(P_{2N}) + 20, time(P_{2N}) + 40]$

Komunikacija

U nekom komunikacijskom sustavu postoje dvije strane: klijent (K) i poslužitelj (P). Uspostava komunikacijskog kanala obavlja se tako da klijent pošalje zahtjev (REQ) na koji poslužitelj odgovara (ACK). Korištenjem Petrijevih mreža modelirati postupak uspostavljanja kanala. Pretpostaviti da se klijent može naći u stanjima:

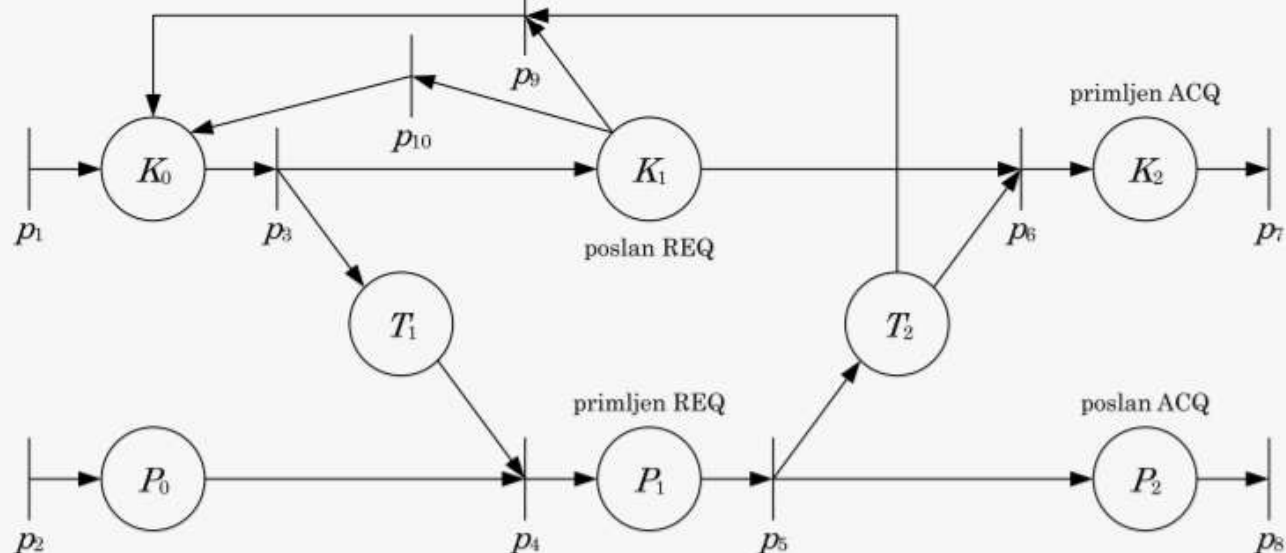
- K_0 – prije slanja zahtjeva REQ (početno mjesto)
- K_1 – nakon slanja zahtjeva REQ, čekanje na odgovor
- K_2 – nakon primitka odgovora ACK – veza uspostavljena.

Poslužitelj se može naći u stanjima:

- P_0 – prije primitka zahtjeva REQ (početno mjesto)
- P_1 – nakon primitka zahtjeva REQ, a prije slanja odgovora ACK
- P_2 – nakon slanja odgovora ACK – veza uspostavljena.

Odgovor na klijentov zahtjev ne smije doći prije $T_1 = 10 \mu s$ (jer za ovaj sustav protok

poruka sigurno traje dulje), ali ni kasnije od $T_2 = 500 ms$ (“timeout”). U oba slučaja klijent treba ponoviti slanje zahtjeva REQ (vratit se u mjesto K_0).



Slika 2.14.

Tablica 2.3.

prijelaz	ulazna mjesta	vremenska ogr
p_1	–	–
p_2	–	–
p_3	K_0	–
p_4	P_0, T_1	–
p_5	P_1	$< time(P_1) + T_1, \infty >$
p_6	K_1, T_2	$< time(K_1) + T_1, time(K_1) + T_2 >$
p_7	K_2	–
p_8	P_2	–
p_9	K_1, T_2	$[0, time(K_1) + T_1]$
p_{10}	K_1	$time(K_1) + T_2$

Prijelazi $p_1 - p_8$ zbivaju se u očekivanom načinu rada, dok se prijelazi p_9 i p_{10} zbivaju u slučaju grešaka.

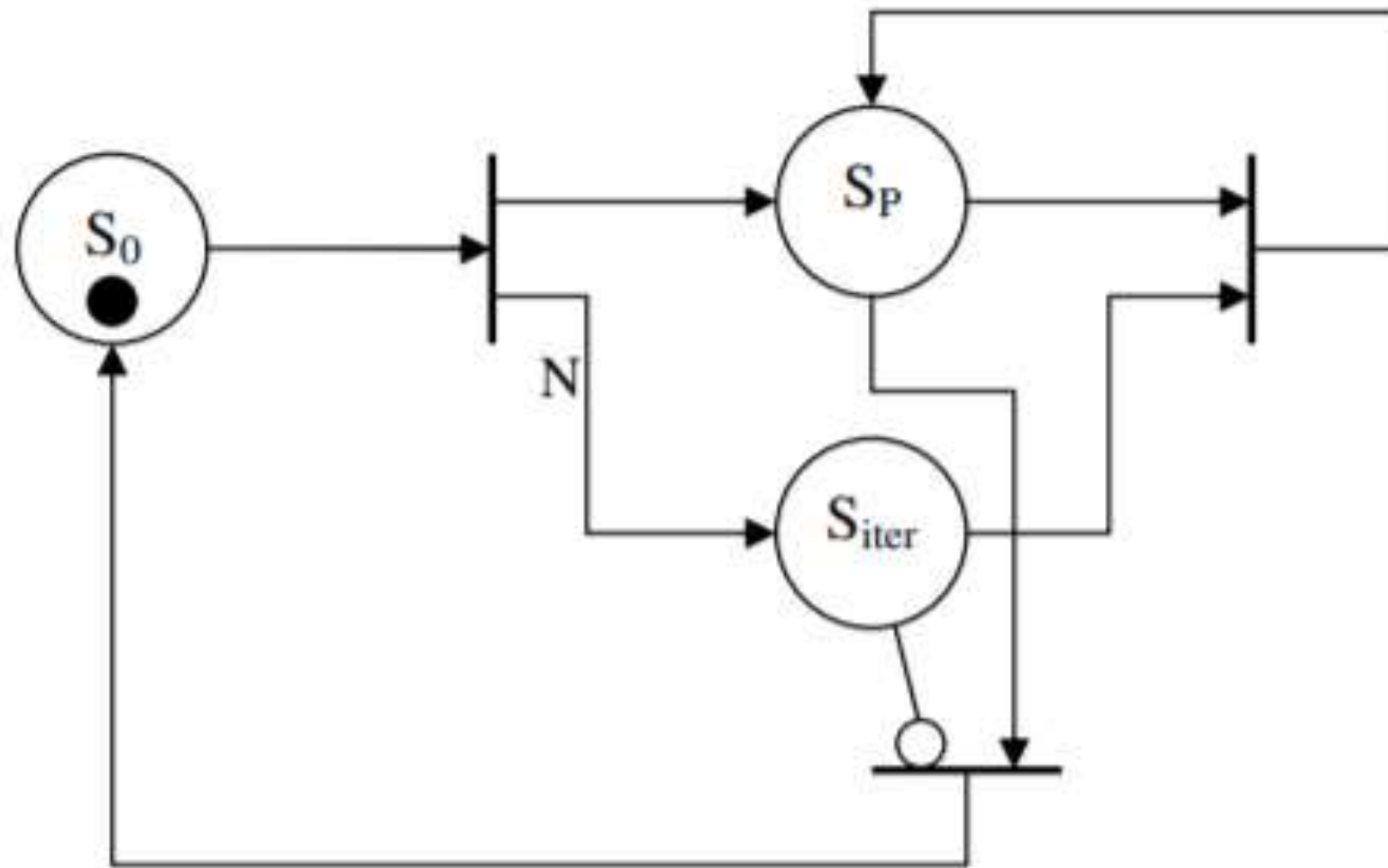
Zadavanje ograničenja na p_6 sprečava nedeterminizam (koji je inače dozvoljen) i moguću grešku, obzirom da bi prijelazi p_6 i p_9 mogli biti omogućeni istovremeno u intervalu $[0, time(K_1) + T_1]$, što bi značilo da su oba događaja ispravna (ili p_6 ili p_9) iako to prema tekstu zadatka ne bi trebalo biti. Slično je i s prijelazima p_6 i p_{10} za trenutak $time(K_1) + T_2$.

zadatak iz blica. nema
rješenja.

5. (0,5) U jednoj Petrijevoj mreži za prijelaz t_x definirano je dodatno vremensko ograničenje tako da sustav ostane u stanju S_x (jedino ulazno stanje u prijelazu t_x , taj je prijelaz jedini iz stanja S_x) od 3 do 5 sekundi. Napisati ograničenje za t_x korištenjem funkcije $time(S)$ koja daje vrijeme pojave tokena u stanju S .

4. (2 boda) Dio svog posla ciklički zadatak obavlja u petlji s N prolaza (izvođenje petlje se ponavlja!). Nacrtati Petri-mrežu koja to prikazuje. Koristiti slijedeće oznake: početno stanje (prije i poslije petlje) je S_0 , a stanje u petlji S_p . Po potrebi dodati i ostala stanja i prijelaze.

2010-2MI



1. U nekom komunikacijskom sustavu postoje dvije strane: klijent (K) i poslužitelj (P). Uspostava komunikacijskog kanala obavlja se tako da klijent pošalje zahtjev (REQ) na koji poslužitelj odgovara (ACK).

(4) Korištenjem petrijevih mreža modelirati postupak uspostavljanja kanala.

Pretpostaviti da se klijent može naći u stanjima:

- K_0 – prije slanja zahtjeva REQ (početno stanje),
- K_1 – nakon slanja zahtjeva REQ, čekanje na odgovor,
- K_2 – nakon primitka odgovora ACK – veza uspostavljena.

Poslužitelj se može naći u stanjima:

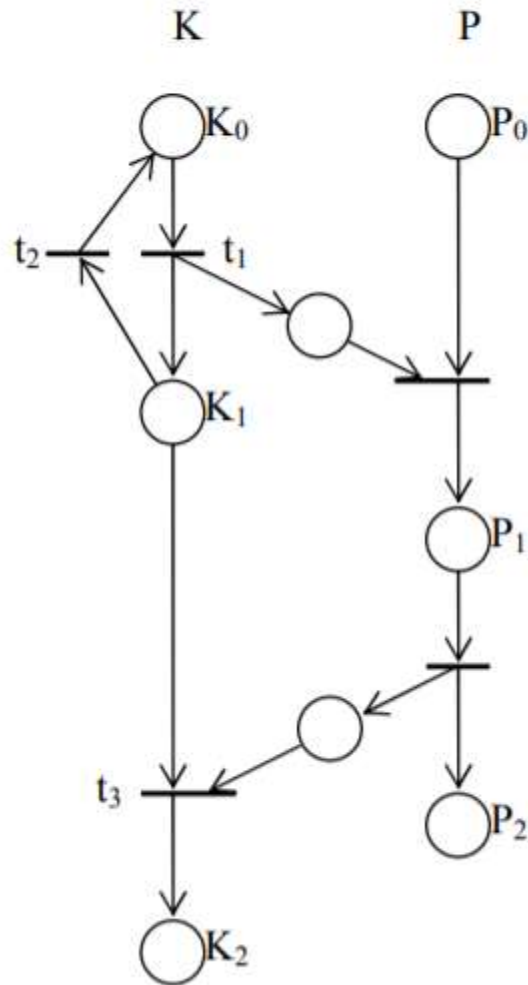
- P_0 – prije primitka zahtjeva REQ (početno stanje),
- P_1 – nakon primitka zahtjeva REQ, a prije slanja odgovora ACK,
- P_2 – nakon slanja odgovora ACK – veza uspostavljena.

Odgovor na klijentov zahtjev ne smije doći prije $T_1=10\ \mu\text{s}$ (jer za ovaj sustav protok poruka sigurno traje dulje), ali ni kasnije od $T_2=500\ \text{ms}$ („timeout“). U oba slučaja klijent treba ponoviti slanje zahtjeva REQ (vratiti se u stanje K_0).

(2) Za klijentsku stranu petrijeve mreže pokazati kada (vremenski gledano) pojedine tranzicije mogu „okinuti“ (napisati tablicu tranzicija koja proširuje petrijevu mrežu u vremensku petrijevu mrežu, ali samo za klijenta).

Rj.

Iduće rješenje je najjednostavnije, sa samo navedenim stanjima za klijenta i poslužitelja te dva „prijenosna“ stanja.



	vrem.	stanja	vremena tranzicija
t1	-	K_0	$(\text{time}(K_0), \infty)$
t2	T_1, T_2	K_1	$(\text{time}(K_1), T_1] \cup [\text{time}(K_1)+T_2, \infty)$
t3	T_1, T_2	K_1	$(\text{time}(K_1)+T_1, \text{time}(K_1)+T_2)$

Osim navedenog, problem je moguće riješiti i bez „prijenosnih“ stanja. Npr. iz t_1 token može ići u P_0 , ali tada veza P_0 do iduće tranzicije mora biti težine 2; također t_2 bi trebao „pakupiti“ jedan token iz P_0 kada se tranzicija dogodi.

Moguće da ima još dobrih rješenja...

6. Neki sustav se početno nalazi u stanju PRAZAN. Nakon prvog događaja AUTO sustav prelazi u stanje JEDAN. Drugi događaj AUTO kad je sustav u stanju JEDAN mijenja stanje sustava u DVA. Kad je sustav u stanju DVA, nakon 1 minute sustav prelazi u stanje PRIJEVOZ. Nakon 10 minuta (u stanju PRIJEVOZ) sustav prelazi u stanje ISKRCAJ. U tom stanju ostaje dok se ne dogode dva događaja ODLAZAK, kada odlazi u stanje PRAZAN. Pretpostaviti da se događaj AUTO neće dogoditi dok sustav nije u stanju PRAZAN ili JEDAN te da će se događaj ODLAZAK dogoditi samo kada je sustav u stanju ISKRCAJ (i to najviše dva takva događaja).

- a) [3 boda] Opisati sustav sekvencijskim dijagramom UML-a za slučaj kada se događaji AUTO događaju u 5. i 15. minuti te događaji ODLAZAK u 30. i 40. minuti.
- b) [3 boda] Prikazati sustav vremenskom Petrijevom mrežom.

6. Neki sustav se početno nalazi u stanju A. Nakon 5 sekundi ili nakon primitka poruke od čvora X prelazi u stanje B. Ako je prijelaz $A \rightarrow B$ uzrokovan porukom, onda u stanju B ostaje 10 sekundi (ako nema novih poruka) te onda prelazi u stanje C. Ako je prijelaz $A \rightarrow B$ uzrokovan protokom 5 sekundi, onda ostaje u stanju B dok ne primi poruku od čvora Y. U stanjima B i C ignoriraju se poruke od čvora X, ali s porukama od čvora Y prelazi se u stanje D. U stanju D ostaje dok ne dođe nova poruka (bilo kakva). Kada dođe poruka, onda sustav prelazi u stanje A.

a) [1 bod] Opisati prikazani sustav UML dijagramom stanja.

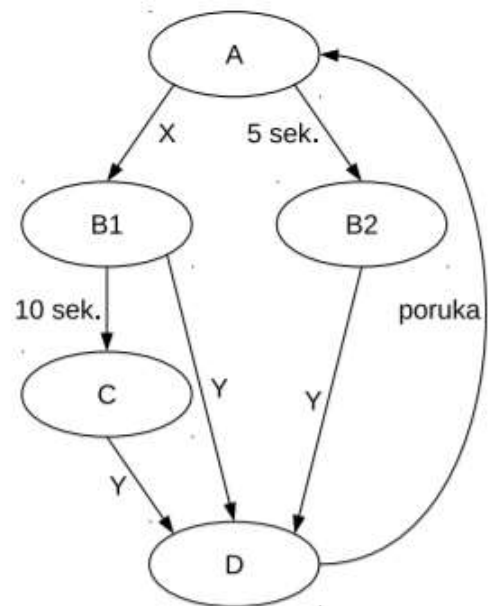
b) [2 boda] Prikazati sustav UML sekvencijskim dijagramom za slučaj kada čvor X pošalje poruku u 3. i 20. sekundi (i nema ostalih poruka).

c) [2 boda] Prikazati sustav vremenskom Petrijevom mrežom.

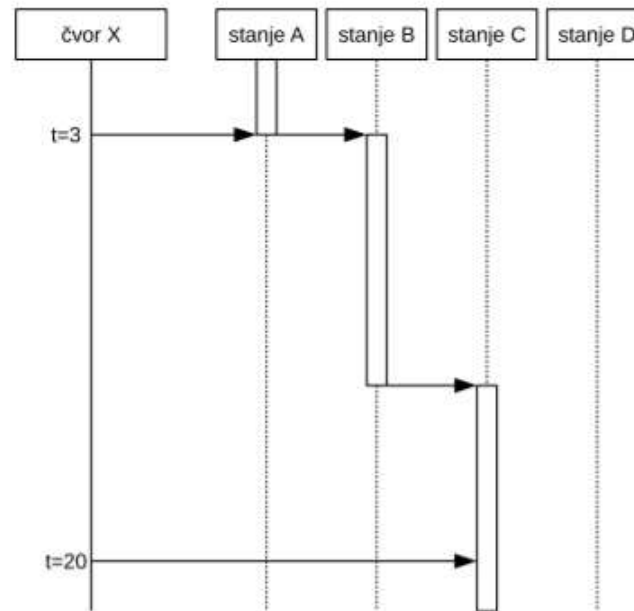
Stanja A, B, C i D su stanja sustava – ne moraju UML dijagrami i Petrijeva mreža imati upravo takva stanja, ali se mora iz njihovih stanja jasno odrediti u kojem je sustav stanju.

2017-MI

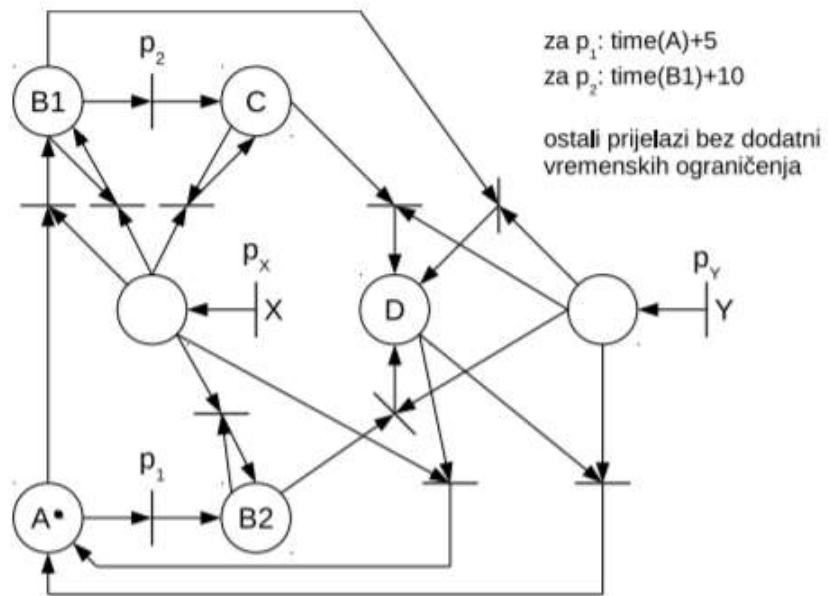
- opet ima rješenja



a) UML dijagram stanja



b) UML sekvencijski dijagram



c) Vremenska Petrijeva mreža

8. Ulazak na parking upravljan je pokretnim vratima. Vrata se otvaraju korištenjem daljinskog upravljača. Uz vrata nalazi se prijemnik koji detektira daljinski te zahtjev proslijeđuje upravljačkom računalu. Ispred vrata, ispod asfalta, nalazi se senzor mase koji također proslijeđuje očitavanja računalu ("ima" ili "nema" auto ispred vrata). Računalo pokreće otvaranje vrata tek pošto je dobio signale i od senzora mase i od prijemnika.
- (a) (2) Prikazati sustav odgovarajućim UML dijagramima (uobičajene operacije).
 - (b) (2) Modelirati otvaranje i zatvaranje vrata vremenskom Petrijevom mrežom uzimajući u obzir da otvaranje i zatvaranje vrata traje 10 sekundi, a da se otvaranje zbiva tek po signalima oba senzora te da zatvaranje kreće tek nakon što senzor mase javi da nema nikog ispred vrata (što uključuje i područje samih vrata).

T1:

Neki sustav se nalazi u stanju A dok ne primi poruku (koja dolazi iz drugog sustava) ili dok ne protekne 10 sekundi. Ako primi poruku prelazi u stanje B, dok po isteku 10 sekundi prelazi u stanje C. U stanju B sustav se zadržava 5 sekundi te se vraća u stanje A. U stanju C se zadržava 10 sekundi prije povratka u stanje A.

4. (4) Opišite sustav opisan s T1 UML dijagramima: sekvencijskim, kolaboracijskim, te dijagramom stanja.
5. (4) Za sustav T1 napravite vremensku Petrijevu mrežu.

- P1. Na jednoj jednosmjernoj cesti nalazi se semafor za pješake i automobile. Semafor ima tipkala T1 za pješake te senzore S1 za automobile i to neposredno prije zebre (na semaforu) te senzore S2 50 metara prije semafora (u smjeru semafora). Ako nema ni pješaka ni automobila stanje semafora je crveno i za pješake i automobile. Kada je „gužva”, tj. ima pješaka i automobila (T1 i S1) onda semafor radi u ciklusu: 45 sekundi zeleno za automobile, 5 sekundi crveno svima, 15 zeleno za pješake, 5 sekundi crveno svima (pretpostavimo da nema žutog). Kada u ciklusu „zeleno za automobile” nema pješaka, a ima automobila (S1) tada se zeleno za automobile produžava (dok je takvo stanje). To produženo stanje se prekida kada se pojavi pješak te se njima pali zeleno na 15 sekundi (prvo automobilima crveno 5 sekundi pa tek onda zeleno pješacima). Vrijedi i obratno, kada nema automobila i ima pješaka tada se njima produžava zeleno, odnosno, pali se zeleno ako je bilo crveno. Ako je sustav u stanju crveno-crveno i detektira se dolazak automobila senzorom S2 tada se zeleno za automobile pali u roku 2 sekunde. Zeleno za pješake uvijek minimalno traje 15 sekundi, a za automobile 45 sekundi.
1. (3) UML dijagramom obrazaca uporabe (*use case*) prikažite operacije semafora i aktore koji u njima sudjeluju (koji ih pokreću) za problem P1. (Voditi se principom *čemu služe ovi dijagrami.*)

2010/11-KZ

- ovaj zadatak ima tri dijela pa ćemo jedan po jedan

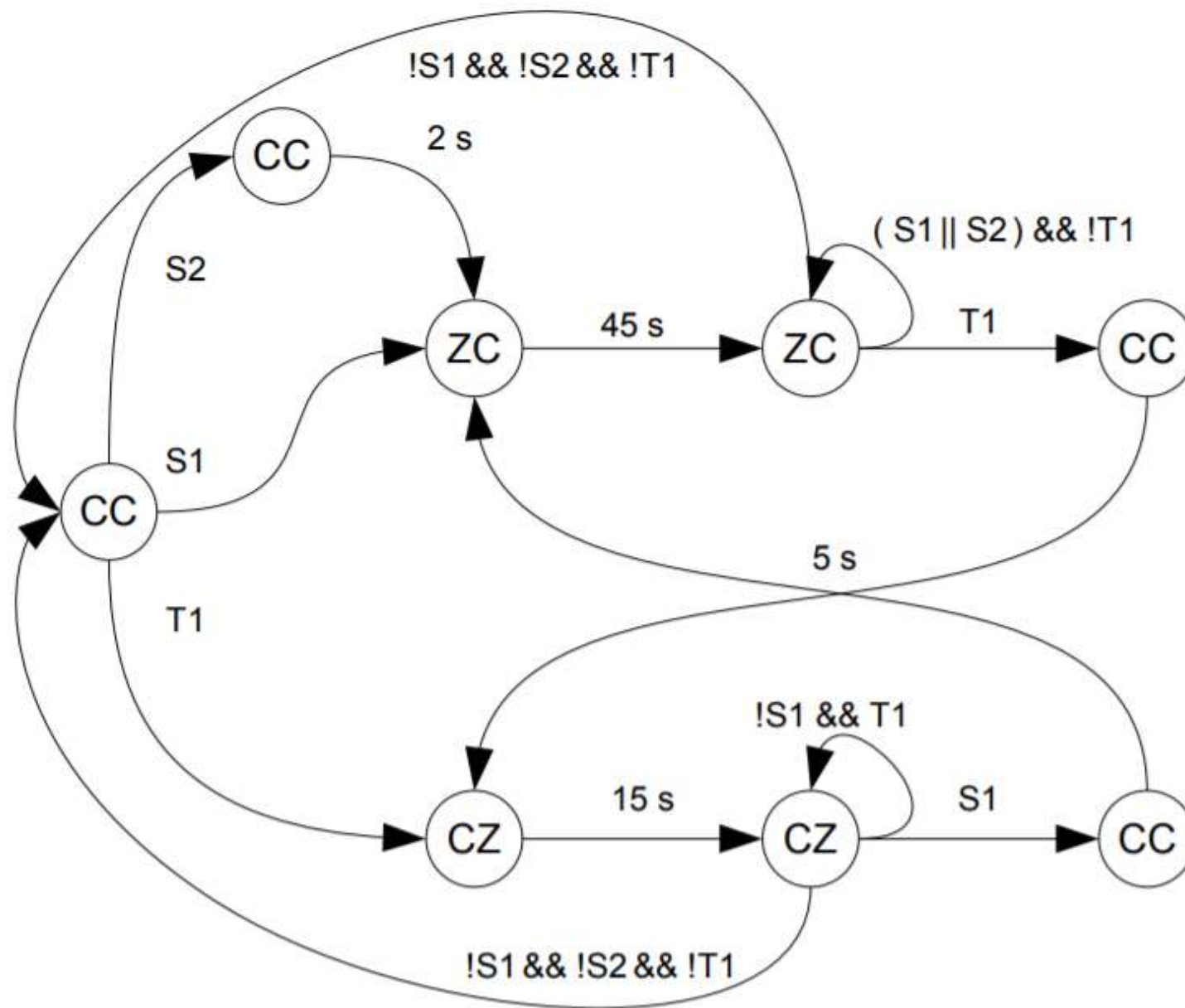
pješake uvijek minimalno traje 15 sekundi, a za automobile 15 sekundi.

1. (3) UML dijagramom obrazaca uporabe (*use case*) prikažite operacije semafora i aktore koji u njima sudjeluju (koji ih pokreću) za problem P1. (Voditi se principom *čemu služe ovi dijagrami*.)

Funkcije koje pruža „sustav” su interakcija preko senzora i tipkala te prikaz svjetala. Te su komponente dio dijagrama zajedno s „aktorima” pješacima i automobilima.

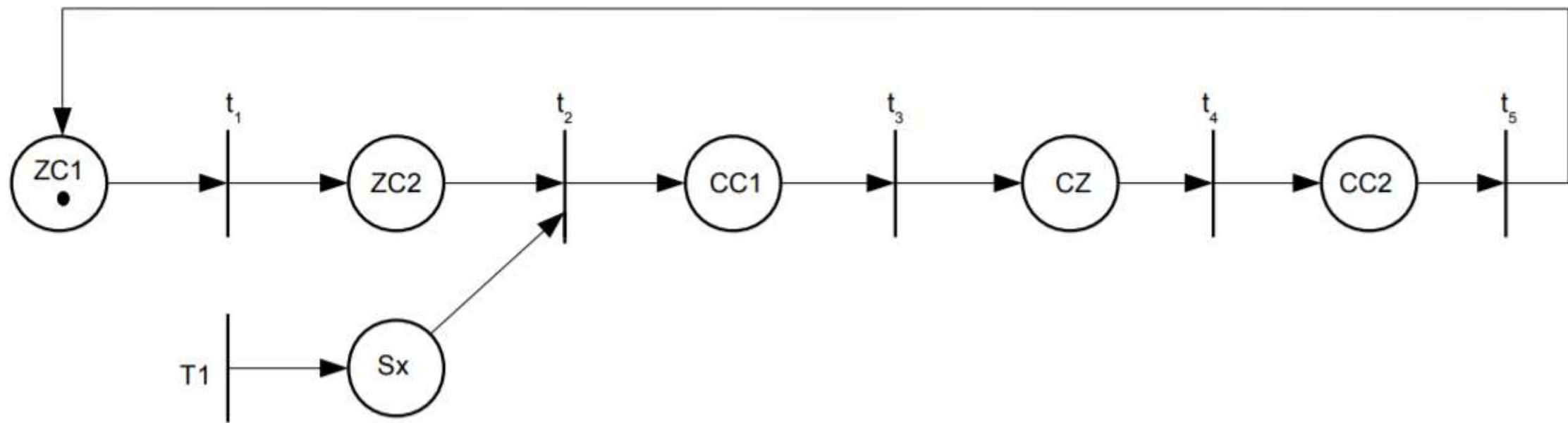
P1. Na jednoj jednosmjernoj cesti nalazi se semafor za pješake i automobile. Semafor ima tipkala T1 za pješake te senzore S1 za automobile i to neposredno prije zebre (na semaforu) te senzore S2 50 metara prije semafora (u smjeru semafora). Ako nema ni pješaka ni automobila stanje semafora je crveno i za pješake i automobile. Kada je „gužva”, tj. ima pješaka i automobila (T1 i S1) onda semafor radi u ciklusu: 45 sekundi zeleno za automobile, 5 sekundi crveno svima, 15 zeleno za pješake, 5 sekundi crveno svima (pretpostavimo da nema žutog). Kada u ciklusu „zeleno za automobile” nema pješaka, a ima automobila (S1) tada se zeleno za automobile produžava (dok je takvo stanje). To produženo stanje se prekida kada se pojavi pješak te se njima pali zeleno na 15 sekundi (prvo automobilima crveno 5 sekundi pa tek onda zeleno pješacima). Vrijedi i obratno, kada nema automobila i ima pješaka tada se njima produžava zeleno, odnosno, pali se zeleno ako je bilo crveno. Ako je sustav u stanju crveno-crveno i detektira se dolazak automobila senzorom S2 tada se zeleno za automobile pali u roku 2 sekunde. Zeleno za pješake uvijek minimalno traje 15 sekundi, a za automobile 45 sekundi.

2. (3) UML dijagramom stanja prikažite stanja semafora te naznačite uzroke prijelaza među stanjima za problem P1.



Pored gornjeg „kompletnog” rješenja priznavana su i jednostavnija, ali ne i ona sa samo tri stanja!

- P1. Na jednoj jednosmjernoj cesti nalazi se semafor za pješake i automobile. Semafor ima tipkala T1 za pješake te senzore S1 za automobile i to neposredno prije zebre (na semaforu) te senzore S2 50 metara prije semafora (u smjeru semafora). Ako nema ni pješaka ni automobila stanje semafora je crveno i za pješake i automobile. Kada je „gužva”, tj. ima pješaka i automobila (T1 i S1) onda semafor radi u ciklusu: 45 sekundi zeleno za automobile, 5 sekundi crveno svima, 15 zeleno za pješake, 5 sekundi crveno svima (pretpostavimo da nema žutog). Kada u ciklusu „zeleno za automobile” nema pješaka, a ima automobila (S1) tada se zeleno za automobile produžava (dok je takvo stanje). To produženo stanje se prekida kada se pojavi pješak te se njima pali zeleno na 15 sekundi (prvo automobilima crveno 5 sekundi pa tek onda zeleno pješacima). Vrijedi i obratno, kada nema automobila i ima pješaka tada se njima produžava zeleno, odnosno, pali se zeleno ako je bilo crveno. Ako je sustav u stanju crveno-crveno i detektira se dolazak automobila senzorom S2 tada se zeleno za automobile pali u roku 2 sekunde. Zeleno za pješake uvijek minimalno traje 15 sekundi, a za automobile 45 sekundi.
3. (6) Sustav sličan problemu P1 nema senzore za automobile već samo tipkala za pješake. On, kad je gužva (kada cijelo vrijeme ima pješaka), radi u istom ciklusu, dok inače drži zeleno za automobile. Kad se pojavi pješak pali crveno za automobile te nakon 5 sekundi zeleno za pješake koje traje 15 sekundi. Nakon toga opet 5 sekundi crveno te minimalno 45 sekundi zeleno za automobile. Modelirati sustav vremenskom Petrijevom mrežom.



$t_1 = \text{time}(ZC1) + 45 \text{ s};$

$t_2 = \max \{ \text{time}(Sx), \text{time}(ZC2) \};$

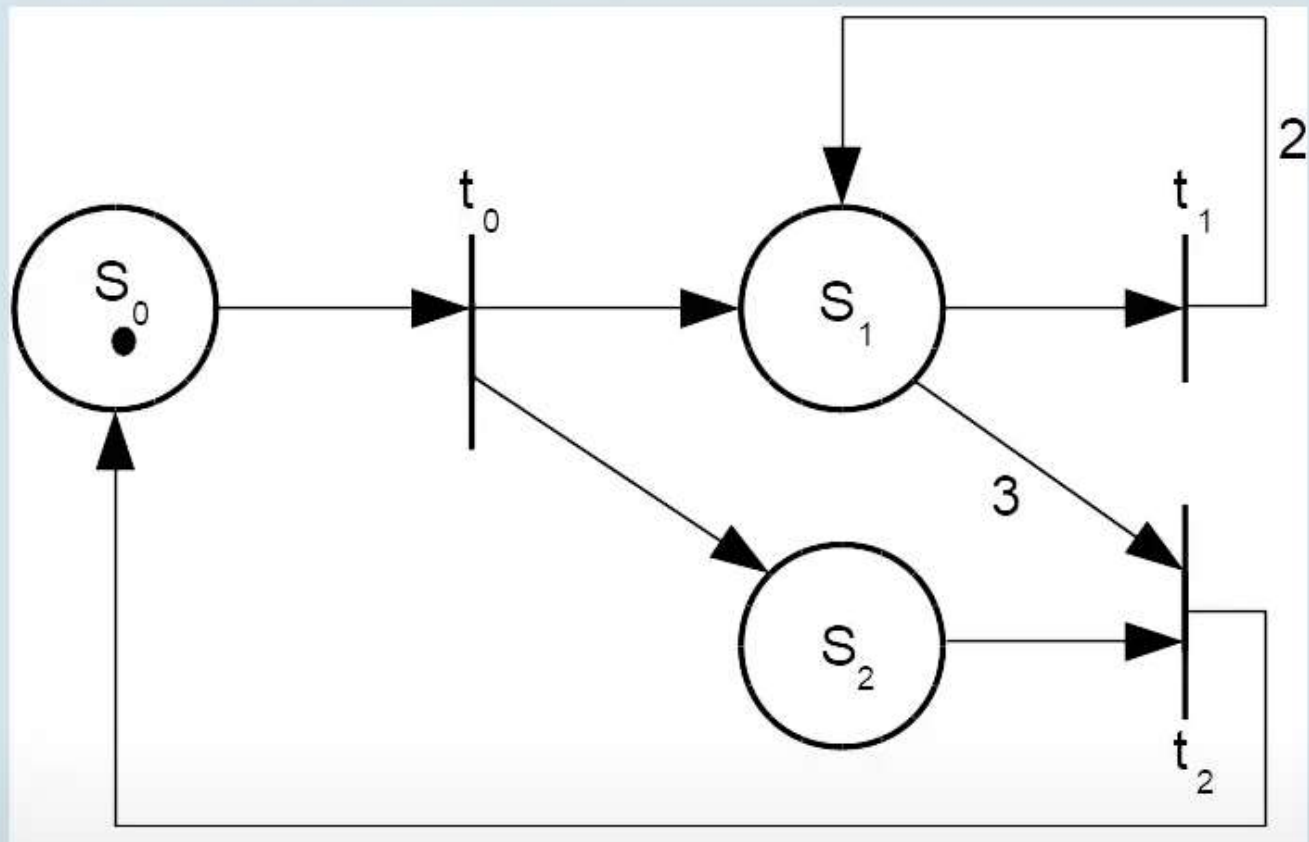
$t_3 = \text{time}(CC1) + 5 \text{ s};$

$t_4 = \text{time}(CZ) + 15 \text{ s};$

$t_5 = \text{time}(CC2) + 5 \text{ s};$

(ili i skraćeno ZC1 i ZC2 u jedan ZC, $t_2 = \max \{ \text{time}(ZC) + 45, \text{time}(Sx) \})$

Zadana je vremenska Petrijeva mreža prema slici:



uz ograničenja:

$t_0: \text{time}(S_0) + 1$

$t_1: \text{time}(S_1) + 5$

$t_2: [\text{time}(S_2) + 2, 100>$

Ako je u $t=0$ situacija kao na slici, kada će ponovno token doći u S_0 ?

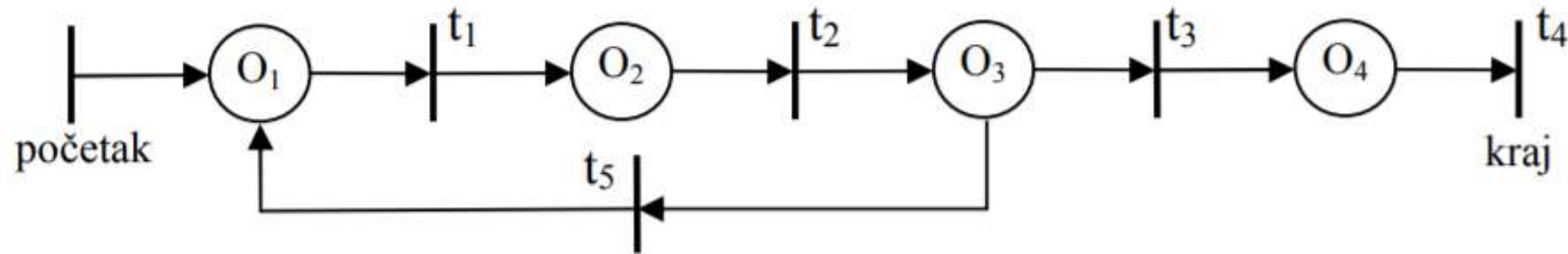
- objašnjenje zadatka iz blica

je na

<https://youtu.be/UAEBiXvP5a0?list=PL5Jyl3Lahn6zxv3beuAwqujuvHcnfrZWF&t=3188>

2010-ZI nema rješenja

1. Neki proces sastoji se od slijedećih operacija: $O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow O_4$. Svaka od operacija traje od 50 do 100 ms. Ponekad (iznimno) operacija O_3 zahtijeva ponavljanje operacija $O_1 \rightarrow O_2$ (tj. jedan niz može izgledati: $O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow O_4$). Petri mreža za zadani proces nacrtana je ispod. Napisati tablicu s vremenima tranzicija.



2009-ZI NEMA RJ

1. (5) U sustavu se nalaze ciklički zadaci dva proizvođača i jednog potrošača. Proizvođači generiraju poruke i stavljaju ih u prazna mjesta međuspremnik (ako je on pun, onda čekaju da se oslobodi jedno mjesto). Potrošač uzima dvije poruke iz međuspremnik te ih procesira (ako nema bar dvije poruke u međuspremniku onda čeka da se poruke pojave). Modelirati sustav *Petrijevom* mrežom. Veličina međuspremnik je 7 poruka.

2014-ZI-NEMA RJ

6. (3) Petrijevom mrežom prikazati komunikaciju jednog procesa proizvođača s tri procesa potrošača. Komunikacija se obavlja preko međuspremniksa sa 5 mjesta za poruke.

13. U nekom sustavu pri obradi jednog zahtjeva sudjeluje pet dretvi: ulazna (koja zaprima zahtjev), tri radne (koje paralelno rade na tom istom zahtjevu te izlazna (koja radi završnu obradu tek kad su radne dretve gotove i vraća rezultat). Prikazati navedeni sustav Petrijevom mrežom.

sljedeći zadaci su
iz zbirke i nemaju
rješenja

14. Sustav s jednim liftom i dva stajanja (prizemlje i prvi kat) ima sljedeća svojstva: otvaranja vrata (i zatvaranja) traje 2 sekunde, vrata su otvorena minimalno 5 sekundi, pomak lifta (na prvi kat, kao i obratno) traje 10 sekundi. Uz pretpostavku samo vanjskih tipki poziva lifta i bez unutarnjih tipki (ako se pozove, lift dođe otvori vrata, zatvori vrata i ode na onu drugu poziciju te otvori vrata), prikazati lift vremenskim Petrijevim mrežama uzimajući u obzir zahtjeve (pozive liftu).

15. Vremenskom Petrijevom mrežom modelirati ulazak kupaca u trgovinu, tj. automatsko otvaranje vrata pred kupcem. Pretpostaviti da se pred pokretnim vratima nalazi senzor koji će detektirati potencijalnog kupca i poslati signal prema upravljačkom računalu koje tada pokreće otvaranje vrata. Otvaranje i zatvaranje traju po dvije sekunde, a minimalno vrijeme s otvorenim vratima jest pet sekundi. Ako senzor i dalje šalje signale o detekciji kupca vrata trebaju biti otvorena za idućih pet sekundi, tj. započet će se sa zatvaranjem najranije pet sekundi nakon zadnjeg signala. Ako se u postupku zatvaranja detektira novi kupac, potrebno je prekinuti zatvaranje i ponovno otvoriti vrata.

16. Neka automatska praonica upravljana je računalom. Sustav se sastoji od tri trake: prve ispred ulaza u praonicu, druge koja vodi auto kroz praonicu do izlaza te treća na izlazu. Vozač se doveze na prvu traku i ugasi auto. Kad auto trenutno u praonici izađe, pokrenu se prva i druga traka, dok novi auto ne uđe u praonicu. Potom počinje pranje. Po dovršetku pranja, ako je prethodno oprani auto na izlaznoj traci otišao, aktiviraju se druga i treća traka i odvezu auto van. Trake su opremljene senzorima težine. Opisati sustav Petrijevom mrežom te UML dijagramom stanja (radi ostvarenja upravljanja).

Još samo teorija

6. Navesti elemente Petrijeve mreže i njihova svojstva.
7. Opisati razliku između vremenskih Petrijevih mreža i običnih Petrijevih mreža. Koje nove mogućnosti nude vremenske mreže?