Pisni del izpita predmeta
OSNOVE MODELIRANJA IN SIMULACIJ
04.02.2000

1.) Z GPSS programskim jezikom opiši naslednji proces: *Na 2 železniška perona prihajajo vlaki iz treh različnih tirov po principu FIFO. Po izstopu ljudi na peronu gre kompozicija v garažo, do katere vodi le en tir.*

2.) Nariši Petrijev graf za proces iz naloge 1.

3.) Do kakšnih označitev pridemo v Petrijevi mreži z začetno označitvijo (1,0,0,0), pri čemer je mreža podana z matrikami:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad O = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

4.) V sistemu M/M/1 se v povprečju nahaja 9 zahtev. Kakšna je verjetnost praznega sistema?

Čas pisanja 50 minut, literatura ni dovoljena. Naloge so enakovredne.

[illegible] proces
FDO. MB

OSNOVE MODULIRANJA IN SIMULACIJ

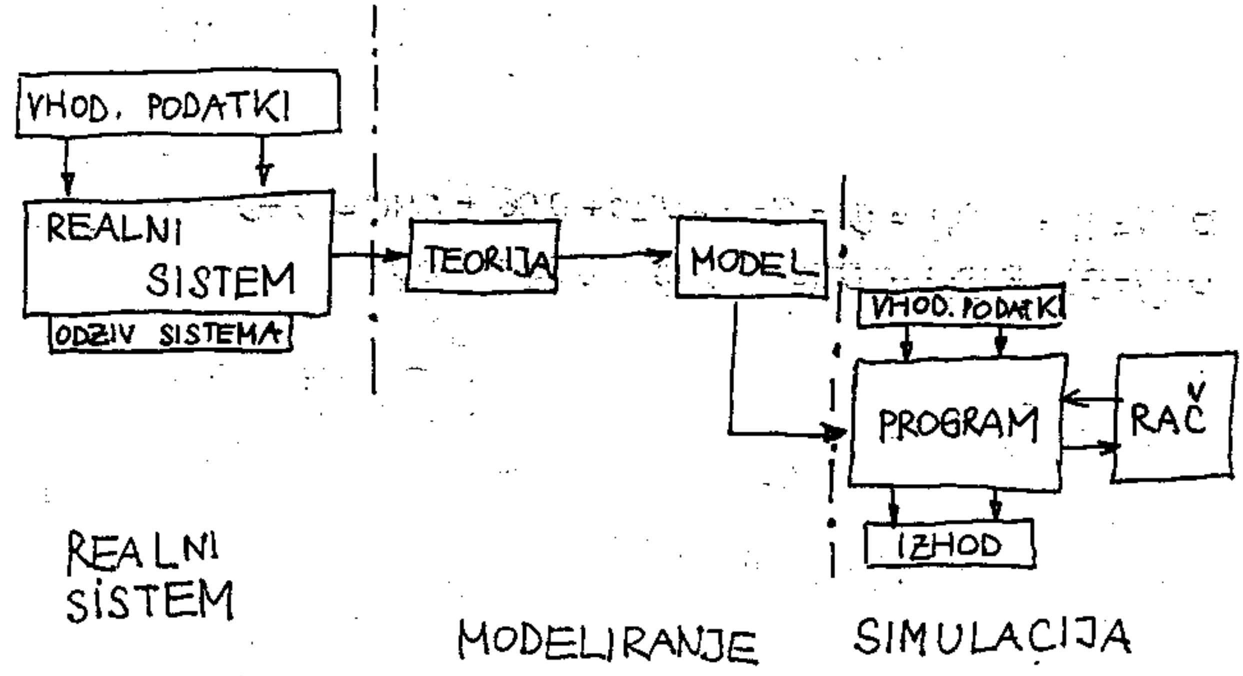
franci. lajovic@kiss.uni-lj.si

REALNI SISTEM - sklop, ki ga simuliramo. Sistem je sestavljen iz komponent, ki morajo med sabo sodelovati

(komponente sistema niso samo HW ampak tudi programi za računalnik)

Vsaka komponenta ima svoje atribute (lastnosti). Opazovanje sistema → opazovanje sistema v določenem stanju (prižgan, ugasnjen)

Sistem je v <u>stacionarnem</u> stanju, kadar je verjetnost določenega stanja konstantna.



Prvi korak pri postavljanju sistema je teorija.

DISKRETNA SIMULACIJA - (vrsta v banki) vse stvari se dogajajo v ločeni časovni obdelavi

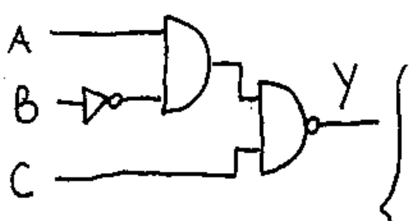
ZVEZNA SIMULACIJA (vezje avtomobila)

<u>RAČUNALNIŠKI SISTEM</u> JE MNOŽICA med seboj povezanih resursov

DETERMINIRANOST - dogodki se dogajajo zaporedno (v enakem intervalu)

VERJETNOST - ('0' se ne zgodi, "1" se vedno zgodi

STOHASTIČNOST - čas glavne spremenljivke, ki vnaša raznolikost v sistem.



$Y = A\bar C \vee \bar B$

dogodki d_i

A B C	FREK.	Y(A,B,C)	$P(d_i)$
0 0 0	0	1	0
0 0 1	18	1	$0{,}30 = \frac{18}{60}$
0 1 0	6	0	$0{,}10 = \frac{6}{60}$
0 1 1	9	0	$0{,}15 = \frac{9}{60}$
1 0 0	15	1	$0{,}25 = \frac{15}{60}$
1 0 1	3	1	$0{,}05 = \frac{3}{60}$
1 1 0	6	1	$0{,}10 = \frac{6}{60}$
1 1 1	3	0	$0{,}05 = \frac{3}{60}$
	$Z = 60$		$\Sigma = 1$

$S = \{\bar A\bar B\bar C, \bar A\bar B C, \bar A B\bar C, \bar A B C, A\bar B\bar C, A\bar B C, AB\bar C, ABC\}$

$P(S) = 1$

$D = \{(\bar A\bar B\bar C, \bar A\bar B C, A\bar B\bar C, A\bar B C, AB\bar C), (\bar A B\bar C, \bar A B C, ABC)\}$

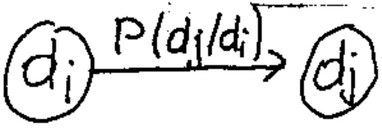
$D = \{D_1, D_0\}$

$P(Y=1) = P(D_1) = 0 + 0'30 + 0'25 + 0'05 + 0'10 = 0'70$

$P(Y=0) = P(D_0) = 0'10 + 0'15 + 0'05 = 0'30$

$P(S) = \sum p(d_i) = 1$

$P(d_j | d_i) = \frac{p(D_i D_j)}{P(D_i)}$



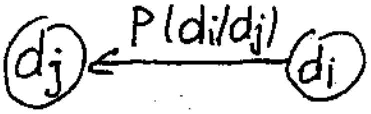
Kakšna je verjetnost d_j po dogodku d_i

$\bar A B \quad P(\bar A B) = P(\bar A B C) + P(\bar A B \bar C) = 0'25$

$\bar A C \quad P(\bar A C) = P(\bar A B C) + P(\bar A \bar B C) = 0'45$

$P(\bar A B/\bar A C) = \frac{P(\bar A C, \bar A B)}{P(\bar A C)} = \frac{P(\bar A B C)}{P(\bar A C)} = \frac{0'15}{0'45} = 0'\bar 3 = \underline{\frac{1}{3}}$

$\boxed{p(d_i/d_j) = \frac{P(D_i D_j)}{P(D_j)}}$



$P(D_i, D_j) = P(D_i/D_j)\,P(D_j) = P(D_j/D_i)\cdot P(D_i)$

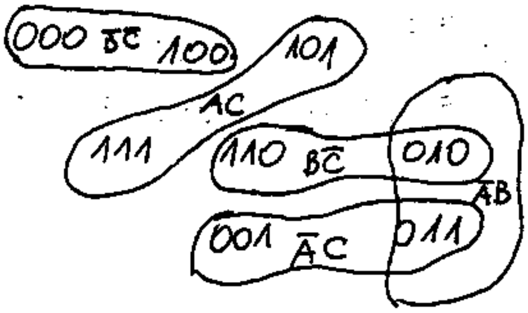
$P(D_j) = \sum_{i=1}^{n} P(D_i \cdot D_j)$ Kakšna je možnost D_j pri vseh predhodnih D_i

$P(D_j) = \sum_{i=1}^{n} P(D_j/D_i)\cdot p(D_i)$

$$\boxed{P(D_i/D_j) = \frac{P(D_j/D_i)\,p(D_i)}{\sum_{i=1}^{n} P(D_j/D_i)P(D_i)}}$$

D_i , D_j nezdružljivi

$$F = \{\bar B\bar C,\ A\bar B,\ AB,\ B\bar C,\ \bar A B,\ \bar A C\}$$



$P(\bar A C / \bar A B)$

$$P(\bar B\bar C) = P(A\bar B\bar C) + P(\bar A\bar B\bar C) = \frac{1}{4} + 0 = \frac{1}{4}$$
$$P(AC) = P(ABC) + P(\bar A BC) = \frac{1}{20} + \frac{1}{20} = \frac{1}{10}$$
$$P(B\bar C) = P(AB\bar C) + P(\bar A B\bar C) = \frac{1}{5}$$
$$P(\bar A C) = \frac{9}{20}$$

$$P(D_j/D_i) = \frac{P(D_i, D_j)}{P(D_i)} = \frac{P(\bar A C/\bar A B)}{P(\bar A B)} = \frac{P(\bar A C, \bar A B)}{P(\bar A B)} = \frac{P(\bar A BC)}{P(\bar A B)} = \frac{\frac{3}{20}}{\frac{1}{4}} = \frac{6}{10} = \underline{\underline{\frac{3}{5}}}$$

$$p(\bar A C/\bar A B) = \frac{P(\bar A B/\bar A C)\,P(\bar A C)}{\sum p(\bar A B/D_i)\cdot P(D_i)} = \frac{\frac{1}{3}\cdot\frac{9}{20}}{\frac{1}{2}\cdot\frac{1}{5} + \frac{1}{3}\cdot\frac{9}{20}} = \underline{\underline{\frac{3}{5}}}$$

$$P(\bar A B/\bar A C) = \frac{P(\bar A B, \bar A C)}{P(\bar A C)} = \frac{1}{3}$$

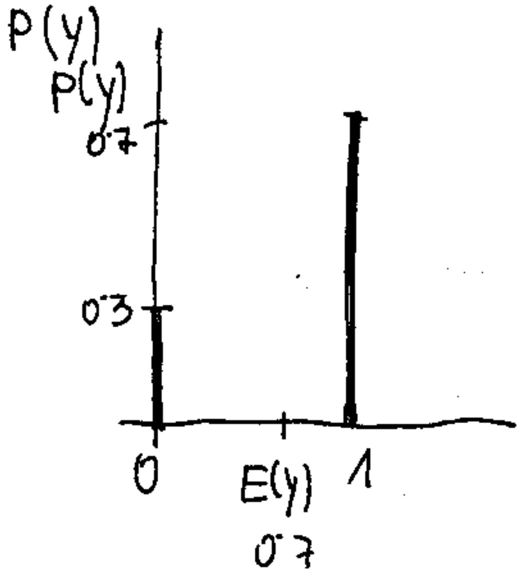
$$\Sigma = P(\bar A B / B\bar C)\ P(B\bar C) + P(\bar A B / \bar A C \mid P(\bar A C)$$

$\bar B\bar C$ $A\bar B$ sta odvisna (statistično)
$A\bar B$ $B\bar C$ sta neodvisna (statistično)

$P(A\bar{B} \vee B\bar{C}) = P(A\bar{B}) + P(B\bar{C}) \nearrow P(A\bar{B}\, B\bar{C}) = 0$

$P(A\bar{B}\bar{C})$

$P(\bar{B}\bar{C} \vee A\bar{B}) = P(\bar{B}\bar{C}) + P(A\bar{B}) \leftarrow P(\overline{BC} \cdot A\bar{B})$



$$E(y) = \sum_{i=1}^{n} y \cdot p_i$$

$$E(y) = P(y=0) \cdot 0 + p(y=1) \cdot 1 =$$

$$= 0{,}3 \cdot 0 + 0{,}7 \cdot 1 = \underline{\underline{0{,}7}}$$

④

A	B	C	Y(A,B,C)	P(d_i)
0	0	0	0	0'20
0	0	1	1	0'10
0	1	0	2	0'05
0	1	1	2	0'10
1	0	0	3	0'20
1	0	1	1	0'05
1	1	0	0	0'10
1	1	1	1	0'20

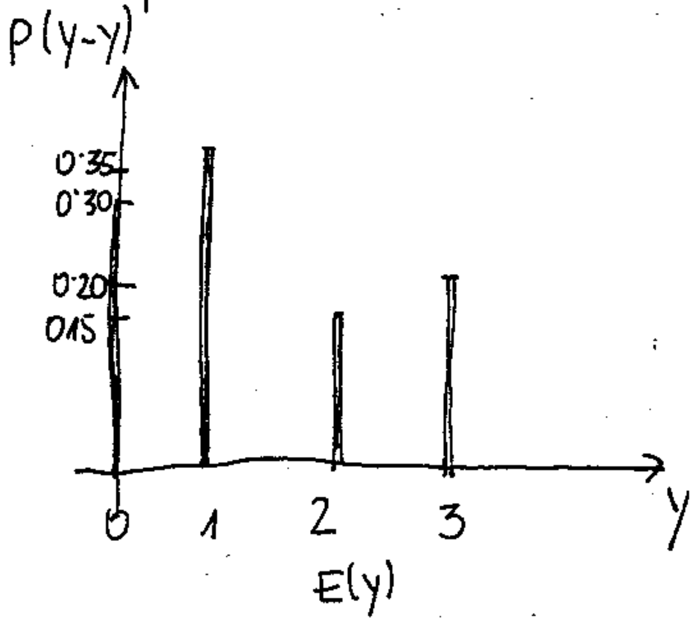
$P(y=0) = p(\bar{A}\bar{B}\bar{C}) + p(AB\bar{C}) =$

$= 0.20 + 0{,}10 = \underline{\underline{0{,}30}}$

$p(y=1) = 0{,}35$

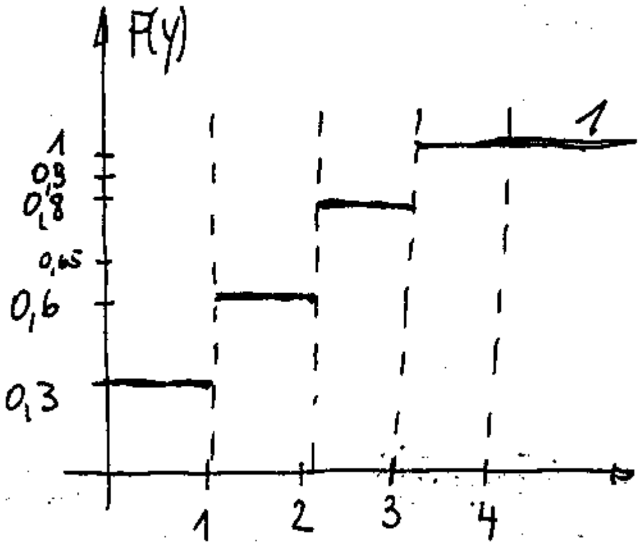
$P(y=2) = 0'15$

$P(y=3) = 0'2$

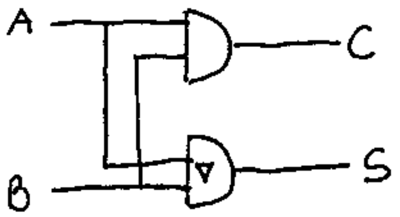


$E(y) = 0,30 \cdot 0 + 0,35 \cdot 1 + 0,15 \cdot 2 + 0,20 \cdot 3 = 1,25$ povprečje

$F(y) = P(Y < y) = \sum_{Y<y} p_i$ porazdelitvene funkcije (gostote verjetnosti)



Primer:



Kakšna je verjetnost izhoda, če tudi za vhod obstaja določena verjetnost ?

$$P_{AC} = P(i_c/v_A)$$

↑

verjetnost, da
bo izhod c
pri vhodu a

Primer matrike

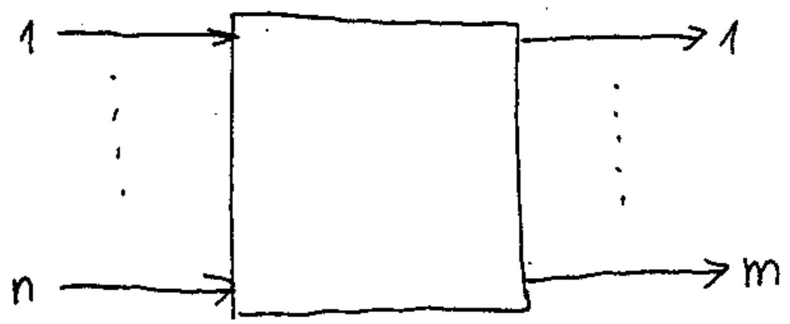
$$M = \begin{array}{cc} & \text{CS} \\ & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ & \left[\begin{array}{cccc} 0{,}50 & 0{,}25 & 0{,}25 & 0{,}00 \\ 0{,}\cancel{12} & \cancel{0{,}80} & 0{,}00 & 0{,}08 \\ 0{,}12 & 0{,}65 & 0{,}00 & 0{,}23 \\ 0{,}00 & 0{,}10 & 0{,}75 & 0{,}15 \end{array}\right] \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} \quad \text{AB} \end{array}$$

Splošno, lahko verjetnost napišemo kot matriko

$$M = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2j} & \cdots & p_{2m} \\ \vdots & & & \vdots & & \vdots \\ \vdots & & & p_{ij} & & \\ p_{n1} & \cdots & & p_{nj} & & p_{nm} \end{bmatrix}$$

$$p_{ij} = P(i_j/v_i)$$

Matriko lahko ponazorimo s shemo



Poglejmo si deterministično matriko primera:

$$M = \begin{array}{cc} & \begin{array}{cccc} 00 & & & \end{array} \leftarrow \text{verjetnost izhoda 00 je 1} \\ \begin{array}{c} 00 \\ \\ \\ \\ \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

← vezje deluje 100% pravilno

Zanima nas kakšna je verjetnost pri izhodu ena

$S = 1$

če

$P(00) = 0{,}10$
$P(01) = 0{,}30$
$P(10) = 0{,}20$
$P(11) = 0{,}40$

↑ verjetnosti vhodov

↑ vsota vseh možnih verjetnosti mora biti 1

običajna pogojna verjetn.

$$p_{ij} = P(i_j \mid v_i)$$

$$P(i_j \mid v_i) = \frac{P(i_j v_i)}{P(v_i)}$$

$$\Downarrow$$

$$p(i_j v_i) = p(i_j \mid v_i) \cdot p(v_i)$$

$$p(i_j) = \sum_{i=1}^{n} p(i_j \mid v_i) \cdot p(v_i)$$

to je v bistvu množenje matrik

verjetnosti pri vhodih

$$V = [p(v_1), p(v_2), p(v_3), p(v_4)] = [0{,}10,\ 0.30,\ 0.20,\ 0.40]$$

enako lahko izpišemo izhodne vektorje

$$I = [p(i_1), p(i_2), p(i_3), p(i_4)]$$

$$\boxed{I = V \times M}$$

Če znamo množiti matrike, bomo znali priti do rezultata.

Primer množenja matrik

$$I = [0{,}10\ 0{,}30\ 0{,}20\ 0{,}40] \begin{bmatrix} 0{,}50 & 0{,}25 & 0{,}25 & 0{,}00 \\ 0{,}12 & 0{,}80 & 0{,}00 & 0{,}08 \\ 0{,}12 & 0{,}65 & 0{,}00 & 0{,}23 \\ 0{,}00 & 0{,}10 & 0{,}75 & 0{,}15 \end{bmatrix} =$$

ⓒⓢ ← skupaj

$$= \begin{array}{cccc} \begin{matrix} 00 \\ c=0 \\ s=0 \end{matrix} & \boxed{\begin{matrix} 01 \\ c=0 \\ S=1 \end{matrix}} & \begin{matrix} 10 \\ c=1 \\ s=0 \end{matrix} & \boxed{\begin{matrix} 11 \\ c=1 \\ S=1 \end{matrix}} \end{array}$$

$$= [0{,}11\ \ 0{,}435\ \ 0{,}325\ \ 0{,}13]$$

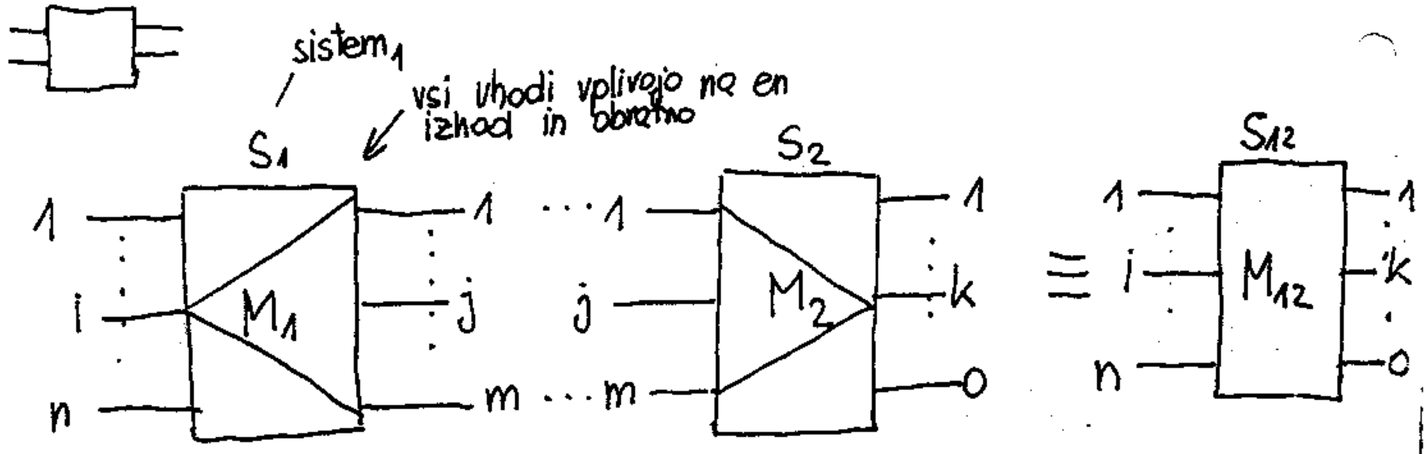
← rezultat so verjetnosti izhodov

nas pa zanima verjetnost, da je $S = 1$

$$\Downarrow$$

$$P(S=1) = P(CS=01,\ CS=11) = P(CS=01) + P(CS=11)$$
$$= 0{,}435 + 0{,}13 = 0{,}565 = \underline{\underline{56{,}5\,\%}}$$

Enako je, če imamo več takih škatel vezano zaporedno



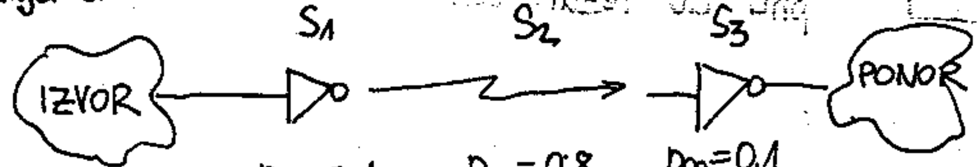
Škatli smo zvezali skupaj, interne povezave se skrijejo.
Primer z logičnim vrati.

Kakšne so relacije v novi matriki M_{12}

$$p_{ik} = \sum_{j=1}^{m} p_{ij}\, p_{jk}$$ kar ni nič drugega kot množenje matrik

$$\boxed{M_{12} = M_1 * M_2}$$

Primer: računanja sistema



$p(0)=0{,}25$	$p_{00}=0{,}1$	$p_{00}=0{\cdot}8$	$p_{00}=0{,}1$
$p(1)=0{,}75$	$p_{01}=0{,}9$	$p_{01}=0{\cdot}2$	$p_{01}=0{,}9$
	$p_{10}=0{,}9$	$p_{10}=0{\cdot}4$	$p_{10}=0{,}9$
	$p_{11}=0{,}1$	$p_{11}=0{\cdot}6$	$p_{11}=0{,}1$
	negator indeksa sta vhod in izhod	prenosna linija	negator

Zanima nas verjetnost, da bo naprava delovala pravilno.
Vsi elementi so podani z matrikami M_{S1}, M_{S2} in M_{S3}

$$M_S = M_{S1} * M_{S2} * M_{S3}$$

matrika celotnega sistema

$$M_S = \begin{bmatrix} 0{,}1 & 0{,}9 \\ 0{,}9 & 0{,}1 \end{bmatrix} \times \begin{bmatrix} 0{,}8 & 0{,}2 \\ 0{,}4 & 0{,}6 \end{bmatrix} \times \begin{bmatrix} 0{,}1 & 0{,}9 \\ 0{,}9 & 0{,}1 \end{bmatrix}$$

$$= \begin{bmatrix} 0{,}557 & 0{,}453 \\ 0{,}292 & 0{,}708 \end{bmatrix} \leftarrow p(1/0)$$

p(0|1)

vnašata napako v sistemu

$$P(\text{napake}) = P(\underset{\text{vhod}}{0}\ \underset{\text{izhod}}{1}) + P(1\ 0)$$

$$= p(1|0)\,p(0) + p(0|1)\,p(1) \doteq \underline{0{,}33}$$

$$P(\text{pravilnega delovanja}) = P(1\ 1) + P(0\ 0) =$$

$$= P(1|1)\,P(1) + P(0|0)\,P(0) \doteq \underline{0{,}67}$$

NAKLJUČNA (SLUČAJNA) SPREMENLJIVKA

je rezultat naključnega procesa (metanje kovanca, kocke). Označimo ga z $\omega \in S$. S je prostor. X je funkcija.

$\omega \in S$

$X(\omega)$

IGRA BLACK JACK. Imamo 3 možnosti, zmago (+5$), vlečemo novo karto ali zgubimo (-5$)

+5	W	(win)	$P[W] = \frac{3}{8}$
0	D	(draw)	$P[D] = \frac{1}{4}$
-5	L	(loose)	$P[L] = \frac{3}{8}$

$$X(\omega) = \begin{cases} +5 & \omega \in W \\ 0 & \omega \in D \\ -5 & \omega \in L \end{cases}$$

BLACK JACK

$$[X = w] = \{\omega : X(\omega) = w\}$$

$$P[X = -5] = \frac{3}{8}$$ ← Verjetnost, da bo vrednost naključne spremenljivke −5

$$P[X = 0] = \frac{1}{4}$$

$$P[\underset{\text{slučajna spremenljivka}}{X} = 5] = \frac{3}{8}$$

PORAZDELITVENA FUNKCIJA

$$[X \leqslant x] = \{\omega : X(\omega) \leqslant x\}$$

$$F_X(x) = P[X \leqslant x]$$

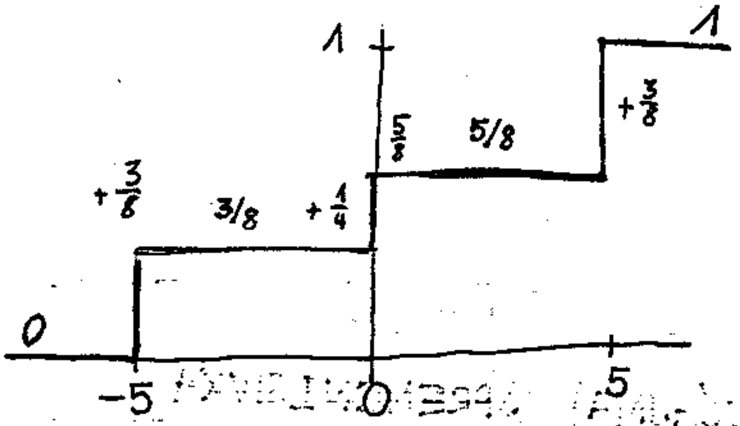
$$F_X(x) \geqslant 0$$

$$F_X(\infty) = 1$$

$$F_X(-\infty) = 0$$

$$F_X(b) - F_X(a) = P[a < x \leqslant b] \quad a<b$$

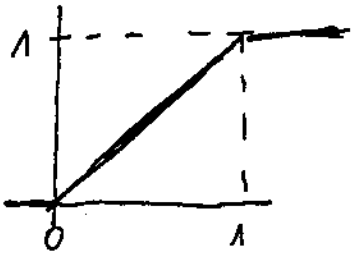
$$F_X(b) \geqslant F_X(a) \quad \text{za} \quad a \leqslant b$$



Verjetnost porazdelitvene funkcije

Iz grafa so razvidne relacije $F_X(\infty)=1$, $F_X(-\infty)=0$ in vse ostale.

Primer: Imamo generator naključnih števil med 0 in 1



Prihod študentov na fakulteto je sicer naključna spremenljivka, saj jo lahko opišemo s porazdelitveno funkcijo.

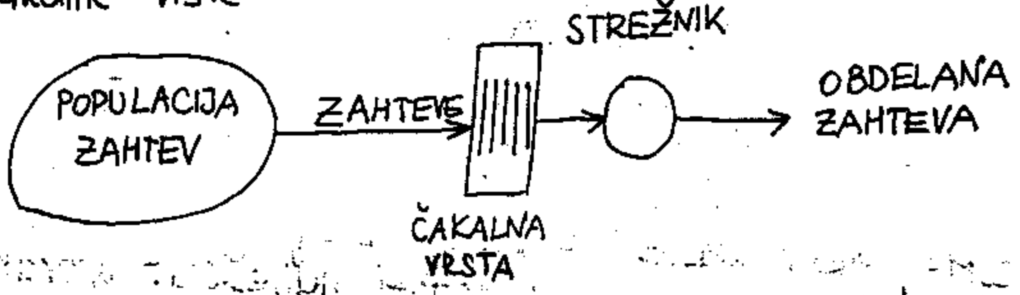
TEORIJA VRST

čakalne vrste

V računalništvu zahteve prihajajo v sistem, se postavijo v vrsto in zapustijo sistem. Podoben primer v banki. Ko vrtimo telefon imamo spet opravka s sistemom, ki se obnaša kot čakalna vrsta.

S tem se ukvarjamo zato, da bo sistem čim bolj izkoriščen - čim hitreje opraviti čim več transakcij. Recimo da imamo dva diska, hitrega in počasnejšega, kako postopati?

Opis čakalne vrste



V splošnem je lahko strežnikov tudi več. Strežnik (čak. vrsto) opišemo s tremi dejavniki

- vhodni proces
- strežni proces
- sistem čakalne vrste

Populacija je zelo pomemben podatek, ko je večja kot 20 zahtev, rečemo da je neskončna in tako imamo opravka z neskončnimi vrstami, ki nas pripeljejo do enostavnega rezultata

Zahteve so lahko deterministične (zelo redki procesi, recimo vsake pet minut nova zahteva). Deterministične zahteve oz. procese je enostavno obravnavati.

Bolj realni pa so naključni procesi. Najbolj običajni procesi so

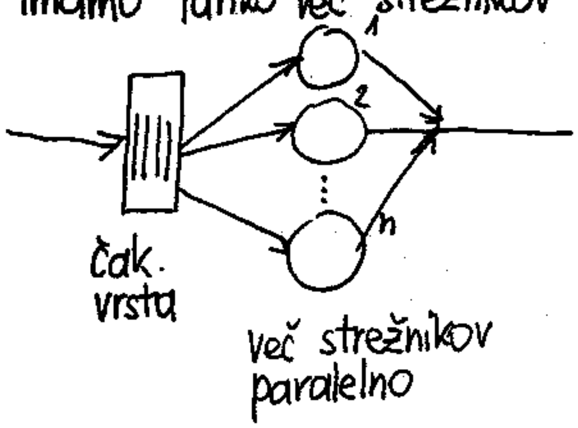
M - najbolj enostaven (poleg det.) in dobro opisuje

D - determinističen proces
zahteve prihajajo v čas. intervalih

G - splošni procesi

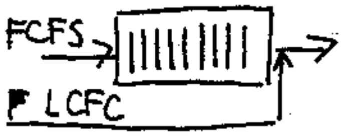
značilnosti zahtev.
Imamo posebne primere, zahteve ki se ne postavijo vrste in zapustijo sistem.

V splošnem imamo lahko več strežnikov

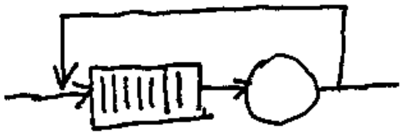


Kapaciteta sistema je maksimalno število zahtev, ki so v strežniku (strežnikih) in čakalni vrsti.

- FIRST COME, FIRST SERVE - najbolj običajna vrsta, ki jih v analizi največkrat uporabljamo
- LAST COME, FIRST COME - primer je sklad



- PRIORITETA, VRSTA S PRIORITETO
- DODELJEVANJE (SHARING) ČASA - večuporabniški sistem procesor delček po delčku obdeluje podatke



- NAKLJUČNO JEMANJE IZ VRSTE

Prioritete so prekinitvene in neprekinitvene. Neprekinitvena počaka spredaj v vrsti, prekinitvena pa primerja prioriteto v strežniku in se izvaja, če je višja.

STR 21 / 19 / 20

STREŽNIK — deterministično D

- verjetnostno s porazdelitvijo M (Poissonova)
- E_k
- splošna strežba

zahteve so posledica naključnega procesa,

$N(t)$ - število zahtev sistema v času t

$N_q(t)$ - število zahtev v čak. vrsti v času t

$N_s(t)$ - število zahtev v strežniku v času t

ŠTEVILO ZAHTEV

$N = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$

$\overline{N}$ - povprečno število zahtev v sistemu

$\overline{N_q}$ - v vrsti $N_q = \frac{\rho^2}{1-\rho}$

$\overline{N_s}$ - v strežniku

POVP. ŠTEVILO ZAHTEV

T_k - čas, ki ga k-ta zahteva porabi v sistem (od vstopa do izstopa iz sistema)

W_k - čas, ki ga k-ta zahteva porabi v čakalni vrsti

X_k - —||— v strežniku

ČASI

$\overline{T}$ POVPREČNI ČASI sistem str 20

$\overline{W}$ čak. vrsta

$\cancel{\overline{W}}$ $\overline{X} = \frac{1}{\mu}$ strežnik

$p_0 = 1-\rho$

$p_v = \rho^v$

$P_k(t)$ - verjetnost, da je v času t k zahtev v sistem

P_k - verjetnost, da je sistem v takšnem stanju (stacionarna verjetnost) za M/M/1 $P_k = (1-\rho)\rho^k$

λ - povp. intenzivnost zahtev, ki pridejo v sistem

μ - —||— , ki jih strežnik obdela na časovno enoto

$\mu = \frac{1}{\overline{x}}$

ρ - uporabnost sistema

$\rho = \frac{\lambda}{\nu} = \lambda \overline{x}$ intenzivnost prometa

$\rho = \frac{\lambda}{m\mu}$ \ št. strežn.

$\overline{N} = \lambda \overline{T}$

$\overline{T} = \overline{W} + \overline{x}$

T

LASTNOSTI

$$N(t) = N_q(t) + N_s(t)$$

$$N = N_q + N_s$$

$$T_K = W_K + X_K$$

$$T = W + \overline{X}$$

povp. vred.

$$N = \lim_{t\to\infty} N(t) = \lim_{t\to\infty} E(N(t))$$

$$N_q = \lim_{t\to\infty} N_q(t) \dots$$

$$T = \lim_{k\to\infty} T_k = \lim_{k\to\infty} E(T_k)$$

$$P_k = \lim_{t\to\infty} P_k(t)$$

Notacija SISTEMA

A/B/X/Y/Z opisujejo strežni sistem

A- verjetnostna porazdelitev na vhodu M- eksponentna porazdelitev

B- -II- strežnega procesa

X - število strežnikov

Y - kapaciteta sistema

Z - disciplina jemanja iz vrste

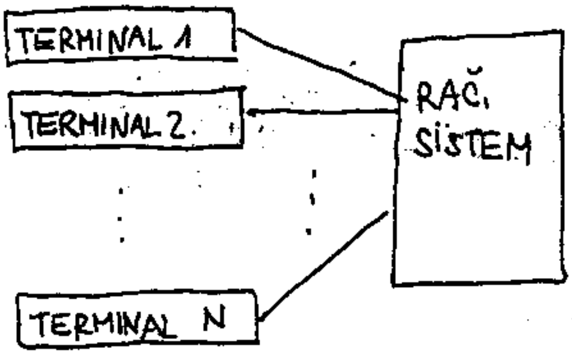
Primer običajnega sistema :

M | M | 1 | ∞ | FCFS , označimo ga z MM1, velikokrat ga uporabimo za modeliranje realnih procesov

M|D|1 – bankomat - hitrost bankomata je vedno enaka (recimo)

Petek, 16. oktober 1998

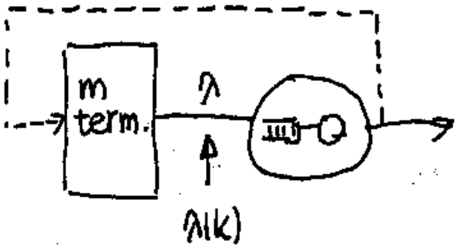
Recimo, da imamo sistem terminalov, ki so povezani v računalniški sistem



Ta sistem si lahko predstavljamo kot

Sistem z omejeno populacijo

Intenzivnost prihajanja zahtev $\lambda(k)$



$$\lambda(k) = \begin{cases} (m-k)\lambda & k \leq m \\ 0 & k \geq m \end{cases}$$

$k \leq m$ - začetna intenzivnost

$k \geq m$ - vsi terminali so dali zahtevo (sistem je s stališča vhoda zaprt)

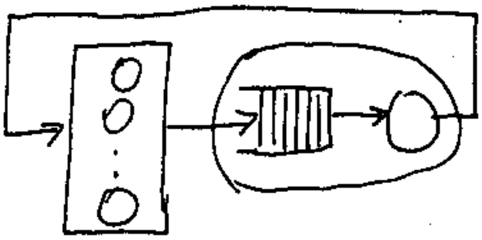
Notacija takega sistema

$\boxed{M/M/1/\infty/m}$

Poissonova porazdelitev

ali kot ~~(pravi)~~ čisti zaprti sistem

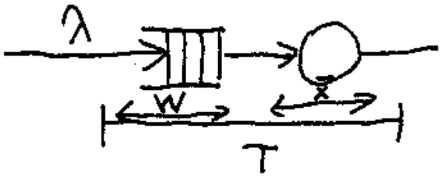
Zahteve krožijo.



LITTL-OV TEOREM

Povp. število zahtev je odvisno od povp. intenzivnosti prihajanja zahtev in povp. časa obdelovanje zahtev.

$N = \lambda T$ Velja za izredno splošne primere, torej ni razlik med vh. porazdelitvami.



ne velja, kadar je intenz. zahtev večja od intenz. strežbe in kadar se zahteve izgubijo v sistemu

$N_q = \lambda W$ Velja za čakalno vrsto

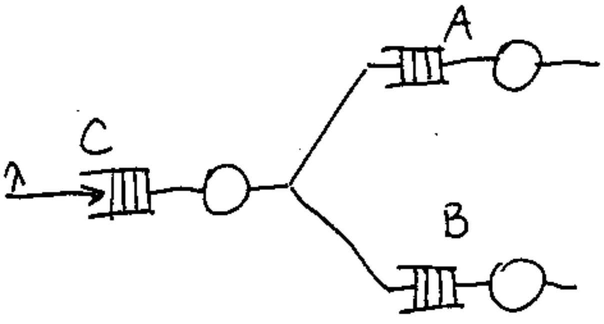
$N_s = \lambda \bar{x}$ Velja tudi za strežnika (strežno enoto)

$N_s = \lambda \bar{x} = \frac{\lambda}{\mu} = \rho$

~~povp. čas~~ strežbe intenzivnost

ρ - faktor uporabnosti

Primer SISTEMA



- Zanima nas število zahtev v sistemu CB

$\underline{N_{CB}} = N_C + N_B =$

$= \lambda T_C + \lambda_B T_B$

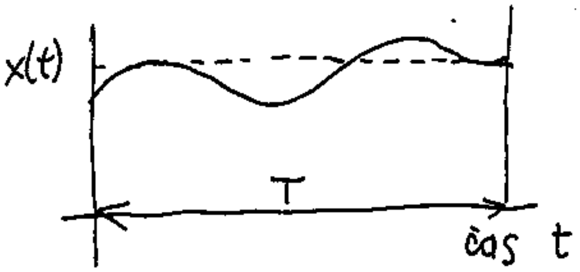
$\underline{T_{CB}} = T_C + T_B$

Če pa zanima celoten sistem:
Littlovo pravilo:

$N = \lambda T$

ERGODIČNOST SISTEMA

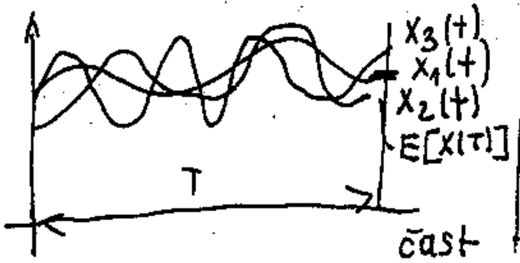
Stohastični proces → naključen proces, v katerem glavno vlogo igra čas



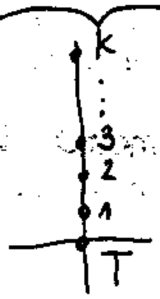
Lahko izračunamo povprečno vrednost $\bar{x}(t)$ ali m

$$\boxed{m = \frac{1}{T}\int_0^T x(t)\,dt}$$

Da pa se tudi drugače. Lahko vzamemo veliko število naključnih spremenljivk $x_1(t)$, $x_2(t)$, $\dots x_n(t)$. Lahko izračunamo povp. vrednost $\bar{x}_i(t)$ ali $E[x(T)]$



$$\boxed{\underbrace{E[X(T)]}_{\text{Povprečje nakl. sprem. v času t}} = \sum_{k=0}^{\infty} k\, \underbrace{P[X(t)=k]}_{\text{verjetnost, da bo nakl. sprem. zavzela vrednost k}}}$$



Vsako od teh lahko zavzame z neko verjetnostjo

Kadar velja

$m = E[X(T)]$, je

SISTEM ERGODIČEN, potem lahko uporabljamo Σ namesto $\int$ in se stvari precej poenostavijo.

FAKTOR UPORABNOSTI, OBREMENITEV RESURSA

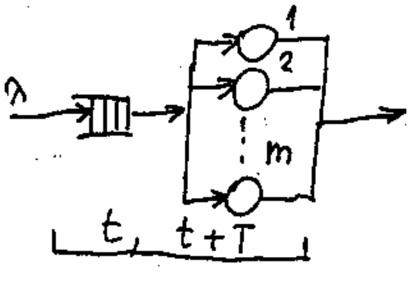
⇓

$$\rho = \frac{\text{ČAS, KO JE STREŽNIK ZASEDEN}}{\text{CELOTNI ČAS}}$$

$0 \leq \rho \leq 1$

sistem je prazen — sistem je 100% zaseden

Primer: Kako izračunamo ρ, če imamo sistem z več strežniki



$$\rho = \frac{\frac{\lambda T}{m} \cdot \bar{X}}{T} = \frac{\lambda}{m\mu}$$

m... število stežnikov

$$\rho = \frac{\lambda}{m\,\mu}$$

μ - intenzivnost strežbe

Koliko zaht/č.e. obdelamo

$\lambda = 10$ ljudi/uro
$\mu = 15$ ljudi/uro
$m = 1$

$$\rho = \frac{\lambda}{m} = \frac{10}{15} = \underline{\underline{66{,}67\,\%}}$$

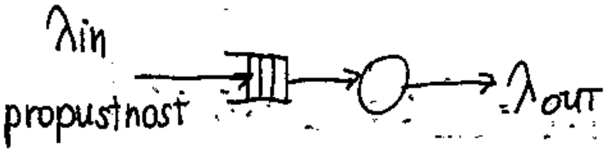
INTENZIVNOST PROMETA (a)

$$a = \lambda \overline{X} = \frac{\lambda}{\mu}$$

Če dobimo rezultat 1.1, potrebujemo dva strežnika (ereklank) erlang

ZAKON O OHRANITVI TOKA

Sistem mora imeti izpolnjene neke zahteve, da je stabilen.



običajno: $\boxed{\lambda_{IN} = \lambda_{OUT}}$ Normalno delujoč, stabilen sistem

$\lambda_{OUT} < \lambda_{IN}$

$\lambda_{OUT} > \lambda_{IN}$

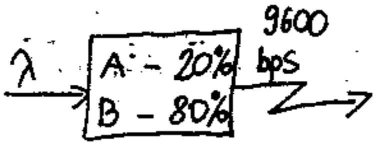
Primer: Imamo komunikacijski kanal, modem, ki deluje 9600 bps. Imamo pakete A (imajo fiksno dolžino 48 bitov), pakete B (povp. dolžine 480 bitov). Zanima nas faktor uporabnosti.

9600 bps

A	48 bit	20%
B	480 bit	80%

$\lambda = 15$ paketov/s

$\rho = ?$



ni pov

(povp. čas, ki ga porabimo na prenosni poti)

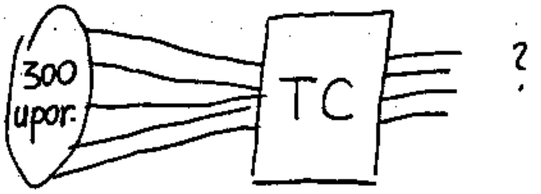
$$\overline{X} = \frac{(0'2 * 48 + 0'8 * 480)}{9600} = 0{,}041 \text{ s}$$

povp. čas strežbo

$$\rho = \frac{\lambda}{\mu} = \lambda \overline{X} = 15 \cdot 0{,}041 \text{ s} = \underline{61{,}5\,\%}$$

Obremenitev sistema je 61,5%

Primer: Imamo telefonsko centralo, na kateri je 300 uporabnikov, vsak opravi povp. 1,5 klica na uro, povp. dolžina klica je 2,5 min. Zanima nas, koliko linij potrebujemo.



1,5 klica/uro
2,5 min

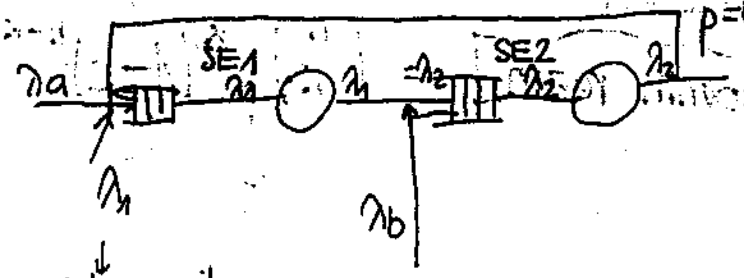
$$d = \underbrace{300 * 1.5}_{} \ \frac{2{,}5}{60} = 18{,}75 \ \text{erlanga}$$

↑da dobimo minute

$$d = \lambda \ \alpha$$

Potrebujemo vsaj 19 linij.

Primer: Imamo povezana dva strežnika, dve strež. enoti, verjetnost povratne poti je 0,5 ← Lahko je to primer na občini



↓ intenz. prih. v prvi strežnik

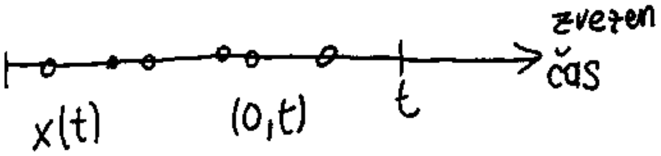
$$\lambda_1 = \lambda_a + 0'5\,\lambda_2 = \lambda_a + 0'5\,(\lambda_1 + \lambda_b)$$

$$\left.\begin{array}{l} \lambda_1 = 2\lambda_a + \lambda_b \\ \lambda_2 = 2(\lambda_b + \lambda_b) \end{array}\right\} \text{s temi podatki potem računamo}$$

POISSONOV PROCES

je verjetnostni proces, ki odlično opiše dogajanje v naravi, naključni proces. Zato je dober model in lahko se z njim račun

↳ je proces, ki nam pove, koliko dogodkov se zgodi v določenem času.



$$P[X(t)=k] = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

distribucijska porazdelitvena funkcija

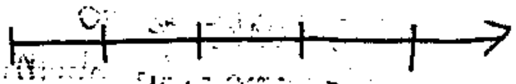
Primer, da se v nekem času ne zgodi noben dogodek

$$P[X(t)=0] = \frac{1}{1} e^{-\lambda t} = e^{\boxed{-\lambda t}}$$

povprečje poissonovega procesa zelo, oz. največji vpliv na verjetnost

BINOMSKA DISTRIBUCIJA, PORAZDELITEV

Na koliko načinov, damo žogice v prostorčke?



diskreten čas

$$P[k \text{ elementov v } n \overbrace{\text{časovnih rezin}}^{\text{diskretna}}] = \binom{n}{k} p^k (1-p)^{n-k}$$

ver. je ne vržemo

vrjetnost, da jo vržemo

Na koliko načinov vržemo dve kroglici

1	2	3	4
O	O		
O		O	
O			O
	O	O	
	O		O
		O	O

$$P[k \text{ prihodov v } \overbrace{(0,t]}^{\text{zvezen čas}}] = \lim_{n\to\infty} \binom{n}{k} \left(\frac{\lambda t}{n}\right)^k \left(1-\frac{\lambda t}{n}\right)^{n-k}$$

$np = \lambda t$ $\qquad p = \frac{\lambda t}{n}$

verjetnost n-tih dogodkov

$$= \frac{(\lambda t)^k}{k!}\left[\lim_{n\to\infty}\frac{n(n-1)\cdots(n-k+1)}{n^k}\right]\left[\lim_{n\to\infty}\left(1-\frac{\lambda t}{n}\right)^n\right]$$

$$= \frac{(\lambda t)^k}{k!}e^{-\lambda t}$$

Čas gledamo kot zvezen proces (Poiss. porazd. je zvezna)

$A(t),\ t \geq 0$

$A(0) = 0$

verjetnost, da zahteva ne pride

funkcija, za katero velja $\lim_{\Delta t \to 0} \frac{O(\Delta t)}{\Delta t} = 0$ vrednost funkcije pada hitreje kot Δt

$$P[A(\Delta t)=0] = 1 - \lambda\Delta t + \underbrace{O(\Delta t)}_{\text{posledica limitnih procesov}}$$

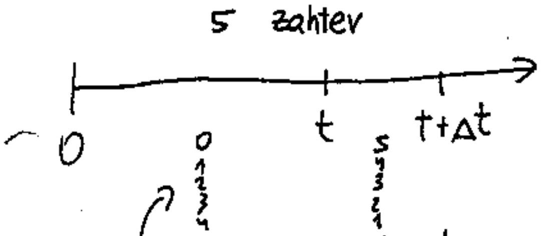
$$P[A(\Delta t)=1] = \lambda\Delta t + O(\Delta t)$$

$$P[A(\Delta t)\geq 2] = O(\Delta t)$$

P_k – v sistemu imamo verjetnost k-tih zahtev

$$P_k(t+\Delta t) = P[k \text{ prihodov v } (0, t+\Delta t]]$$

$$= \sum_{i=0}^{k} P[(k-i) \text{ v času } (0,t) \text{ in } i \text{ v času } \Delta t]$$



verjetnost, da pride do takih razporeditev

$$= \sum_{i=1}^{k} P[(k-1) \text{ v času } (0,t]] \times P[i \text{ v času } \Delta t]$$

$$= \underbrace{P_k(t)}_{k \text{ zahtev}} \underbrace{[1-\lambda\Delta t + O(\Delta t)]}_{\text{ne pride nobena zahteva več}} + \underbrace{P_{k-1}(t)}_{k-1 \text{ zahtev}} \underbrace{[\lambda\Delta t + O(\Delta t)]}_{\text{pride še ena zahteva}}$$

$\Delta t \to 0$

$$\frac{dP_k(t)}{dt} = -\lambda P_k(t) + \lambda P_{k-1}(t) \quad k>0$$ dif. enačba

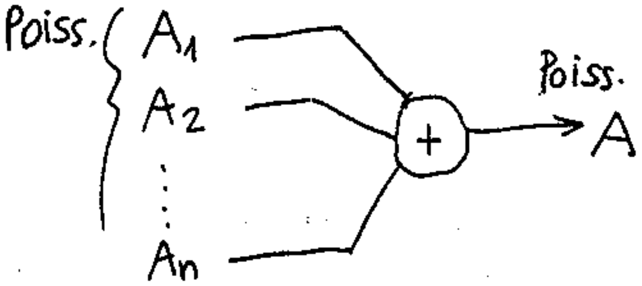
Začetni pogoj: $\frac{d P_0(t)}{d(t)} = -\lambda P_0(t)$

$P_0(0) = e^{-\lambda t}$

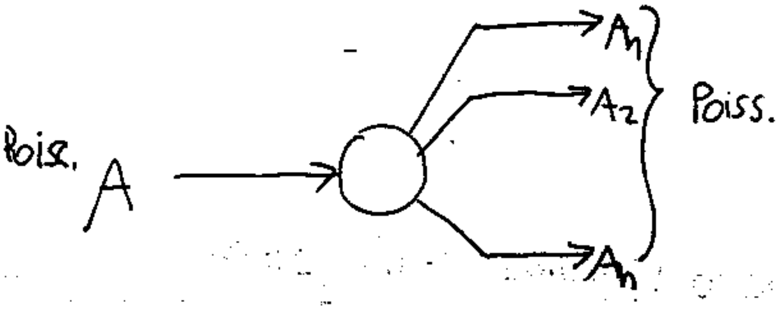
$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$

ZNAČILNOSTI POISS. PROCESA

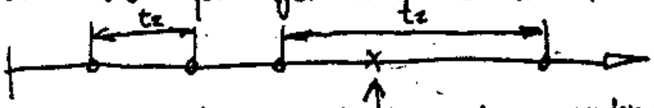
Če zlivamo poissonove procese (jih seštevamo), bo rezultat ponovno poissonov proces.



Velja obratno. Imamo poissonov proces, ki se nam razdeli na poissonove procese



Ta Poissonov proces je brez pomnenja. Ne glede, kdaj se je dogodek zgodil, je verjetnost naslednjega dogodka enaka. Primer z avtobusom. (Če prihaja avtobus na 10 minut)

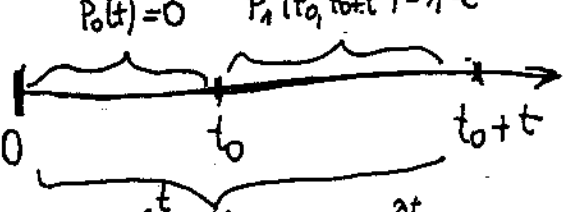


Za opazovalca zgodovina ni pomembna.
PROCES BREZ POMNENJA

$$P[\text{ni prihodov v } (0,t_0)] = \underline{\underline{e^{-\lambda t_0}}}$$

$$P[1 \text{ prihod } (t_0,t_0+\Delta t) \,/\, \text{ni prihodov v času } (0,t_0)] =$$

$$= \frac{\int_{t_0}^{t_0+\Delta t} \lambda e^{-\lambda t}\, dt}{e^{-\lambda t_0}} = \underline{\underline{1-e^{-\lambda t}}}$$



$$P_1(0,t)=\int_0^t \lambda e^{-\lambda t}\,dt = 1-e^{-\lambda t}$$

zgodovina ne vpliva na verjetnost dogodka

$N = \lambda T$ STOHASTIČNI PROCES

$\{x(t), t \in T\}$ $P[x(t) = i]$

$\uparrow$ naključna spremenljivka

verjetnost, da je naša naključna spremenljivka zavzela vrednost i

PROSTOR STANJ

$x(t) = i$ proces je v času t v stanju i
i - celo število (diskreten prostor stanj)
obstajajo tudi zvezni prostori stanj
↓
običajno bodo to števila $\{0, 1, 2, \ldots\}$

INDEKSNI PARAMETER

t - čas je v glavnem indeksni parameter (v stohastičnih procesih)

DISKRETNI (v točno določenih intervalih)
ZVEZNI (kadarkoli) ← prehod

PROSTOR STANJ

INDEKSNI PAR.	DISKRETNI	ZVENZNI
DISKRETNI	diskretna časovna stohastična veriga	diskretno časovni stohastični proces
ZVEZNI	zvezna časovna stohastična veriga	zvezni časovni stohastični proces

Obravnavali bomo diskretno časovno stohastično verigo in zvezno časovno stohastično verigo.

Primer: BORZA

$\{X_n\}$

X_K - časovni korak ← PRIMER diskretne časovne stoh. verige

$X_K = 2{,}45$ ↖ (ko se borza zapre, velja tečaj do naslednjega dne
$X_{K+1} = 2{,}38$ to je diskretno stanje (če je npr 2,45 SIT)
$X_{K+2} = 2{,}29$
$X_{K+3} = 2{,}78$ ↙ Primer zvezne časovne stohastične verige

$t_1 < t_2 < t_3 < t_3$ $\{X(t)\}$

$X(t_1) = 2{,}38$ $X(t_3) = 2{,}40$
$X(t_2) = 2{,}39$ $X(t_4) = 2{,}36$

DISKRETNE ČASOVNE MARKOVSKE VERIGE

Markovski proces je stohastični.
Primer Markovske verige, → ko nimamo zgodovine.

<u>Primer</u>: Mečemo kovanec, cifra ali mož (C/M)

$X_K = 1$ cifra $X_K = 1$ - v k-tem poskusu smo vrgli cifro

Y_k = "vsota metov cifre"
$Y_0 = 0$ - začetno stanje

	meti									
	C	C	M	M	C	M	C	C	C	M
X_k	1	1	0	0	1	0	1	1	1	0
Y_k	1	2	2	2	3	3	4	5	6	6

vidimo:

$$Y_K = \underbrace{Y_{K-1}}_{\text{trenutno stanje}} + \underbrace{X_K}_{\text{kar smo vrgli}}$$

$\{X_k, k \in T\}$ — zanima nas verjetnost, da iz stanja i pridemo v stanje j v odvisnosti od časa

$$P[X_{k+1} = j \mid X_0 = i_0, X_1 = i_1 \dots X_{k-1} = i_{k-1}, X_k = i_k]$$

v tem primeru imamo vso zgodovino
Splošno lahko napišemo

$$= P[X_{k+1} = j \mid X_k = i] = P_{ij}$$

Verjetnost, da iz stanja i pridemo v stanje j (zgodovina ni pomembna)

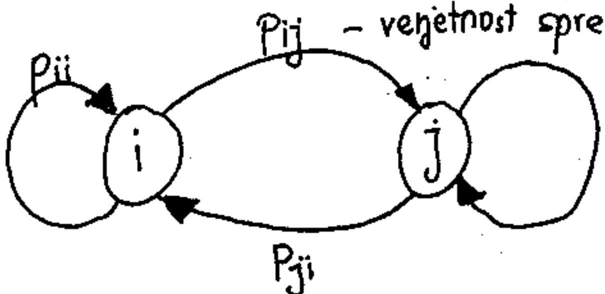
VERJETNOST PREHAJANJA (PREHODA)

ČASOVNO HOMOGENE MARKOVSKE VERIGE
v verjetnosti P_{ij} čas ne nastopa,
P_{ij} ni odvisna od časa.

$X_K = i$ ← prostor stanj

V Markovskih verigah bo prostor stanj vedno celo število
$\{0, 1, 2, . \}$ $X_k = i$) – proces je v času X_K enak i.

$P[X_k = j] \equiv \pi_j^{(k)}$ π - pomeni, ver: da smo v čas. trenutku k v stanju j, verjetnost



p_{ij} - verjetnost spremembe v enem koraku (časovnem)

p_{ij} - diskretni čas

$k = 0, 1, 2, 3, \ldots$ v diskretnih Markovskih verigah čas jemljemo kot števila

Zanima nas začetno stanje, kje začnemo?

$P[X_0 = i]$ - začetna verjetnostna porazdelitev

Zanima nas $P[X_{k+1} = j]$ ← Kakšna je verjetnost, da bomo v naslednjem časovnem trenutku v stanju j

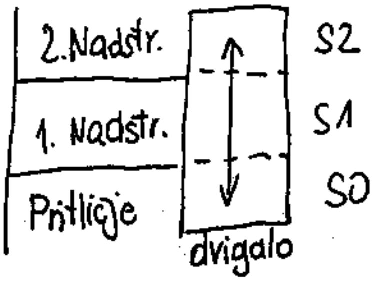
$$P[X_{k+1} = j] = \sum_{i=0}^{\infty} P[X_k = i] \cdot P[X_{k+1} = j \mid X_k = i]$$

je vsota (vseh možnih izhodišč × verjetnost da iz tega stanja preidemo v drugo stanje)

$$= \underline{\underline{\sum_{i=0}^{\infty} \pi_i^{(k)} \cdot p_{ij}}}$$

← ta verjetnost se ne spreminja po času (Č.H. MAR. VERIGA)

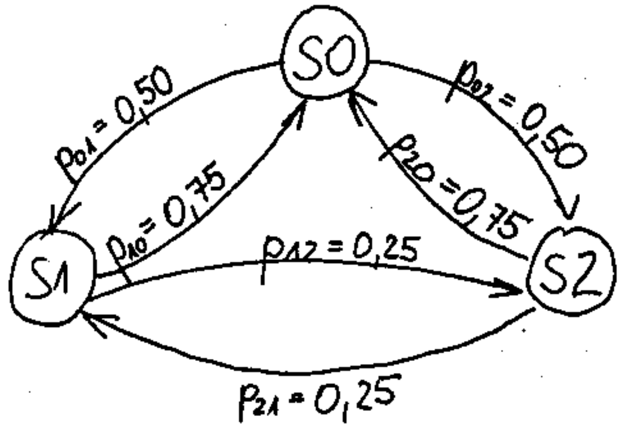
<u>Primer</u>: Imamo dvigalo, ki lahko ima tri stanja: S0, S1 in S2. Zanima nas verjetnosti. Tabela prehodov.



Verjetnosti nasl. stanja

Trenutno stanje	Naslednje stanje S0	S1	S2
S0	0	0,50	0,50
S1	0,75	0	0,25
S2	0,75	0,25	0

Čas je diskreten, ker je pogojen s premikom dvigala iz enega v drugo nadstropje. Lahko si narišemo diagram prehajanja stanj.



Zanima nas začetno stanje

$\pi_0^{(0)} \equiv P[X_0 = 0] = 1$ – v začetku smo v pritličju

$\pi_1^{(0)} \equiv P[X_0 = 1] = 0$

$\pi_2^{(0)} \equiv P[X_0 = 2] = 0$ X_0 – pove v katerem stanju smo v stanju ∅

Kakšna je verjetnost za posamezen prehod v naslednjem časovnem koraku

$$\pi_0^{1} = P[X_1 = 0] = \sum_i P[X_1 = 0 \mid X_0 = i] \cdot P[X_0 = i] =$$

$$= \pi_0^{0} \cdot p_{00} + \pi_1^{0} \cdot p_{10} + \pi_2^{0} \cdot p_{20} = \underline{\underline{0}}$$

$$(1 \cdot 0 + 0,75 \cdot 0 + 0 \cdot 0,75)$$

verjetnost, da sem v času nič v stanju nič

ver. prehodov

$$\pi_1^{1} = P[X_1 = 1] = \pi_0^{0} p_{01} + \pi_1^{0} p_{11} + \pi_2^{0} p_{21} =$$

$$= 1 \cdot 0,5 + 0 \cdot 0 + 0 \cdot 0,25 = \underline{\underline{0,5}}$$

Podobno dobimo $\pi_2^{1} = \underline{\underline{0,5}}$.

$$\pi_0^{2} \equiv P[X_2 = 0] = \pi_0^{1} p_{00} + \pi_1^{1} p_{10} + \pi_2^{1} p_{20} =$$

$$= 0 \cdot 0 + 0,5 \cdot 0,75 + 0,5 \cdot 0,75 = \underline{\underline{0,75}}$$

zrjetnost tanja ∅ po lveh prehodih

, podobno dobimo:

$\pi_1^{2} = 0,125$

$\pi_2^{2} = 0,125$

Uporabili bomo matriko vseh prehodov

$\boxed{\tilde{P} = (p_{ij})}$ posamezni členi matrike so verjetnosti prehodov iz i v j stanje

$\tilde{\Pi}^{(K)} = (\Pi_0^K, \Pi_1^K, \Pi_2^K \dots \Pi_n^K)$ vektor, ki združuje vsa stanja

Vidimo, da velja:

$$\left.\begin{aligned} \tilde{\Pi}^{(1)} &= \tilde{\Pi}^{(0)} \tilde{P} \\ \Pi^{(2)} &= \tilde{\Pi}^{(1)} \tilde{P} \\ &\vdots \\ \tilde{\Pi}^{(K)} &= \tilde{\Pi}^{(K-1)} \tilde{P} \end{aligned}\right\}$$

Lahko pa $\tilde{P}$ izvzamemo

$$\boxed{\tilde{\Pi}^{(K)} = \tilde{\Pi}^{(0)} \cdot \tilde{P}^{(K)}}$$

$P^{(0)} = I$ - enotna matrika

$$\tilde{P}^{(k+1)} = \tilde{P}^{(k)} \cdot \tilde{P}$$

$$\boxed{P_{ij}^{k+1} = \sum_{l=0}^{n} P_{il}^{k} \cdot P_{lj}}$$

Chartmanova enačba

verjetnost, da pridemo iz točke i do j v k+1 korakih

$$\tilde{P} = \begin{bmatrix} 0 & 0,5 & 0,5 \\ 0,75 & 0 & 0,25 \\ 0,75 & 0,25 & 0 \end{bmatrix}$$

Vsota po vrsticah je enaka 1

$$\tilde{P}^{(2)} = \tilde{P} \cdot \tilde{P} = \begin{bmatrix} 0,75 & 0,125 & 0,125 \\ 0,1875 & 0,4375 & 0,375 \\ 0,1875 & 0,375 & 0,4375 \end{bmatrix}$$

↓ verjetnost prehoda v dveh korakih

$\tilde{\Pi}^{(0)} = [1, 0, 0]$

$\tilde{\Pi}^{(1)} = \tilde{\Pi}^{(0)} \times \tilde{P} = [0, 0'5, 0'5]$

$\tilde{\Pi}^{(2)} = \tilde{\Pi}^{(1)} \times \tilde{P} =$

$= \tilde{\Pi}^{(0)} \tilde{P}^{(2)} =$

$= [0'75, 0'125, 0'125]$

Stacionarne značilnosti Markovskih verig

če pri dvigalu ponavljamo prehode v neskončnost, je posamezna verjetnost prehodov fiksna.

$$\lim_{k\to\infty} P_{ij}^{(k)} = \Pi_j$$

$$\lim_{k\to\infty} \Pi_j^{(k)} = \sum_i \tilde{\Pi}_i^{(0)} P_{ij}^{(k)} = \Pi_j \underbrace{\sum_i \Pi_i^{(0)}}_{1} = \underline{\underline{\Pi_j}}$$

↓ verjetnost določenega stanja po $k\to\infty$ postane stacionarna.

Stac. stanja so zanimiva, ker lahko pri čakalnih vrstah lahko izračunamo povprečne dolžine vrst.

↑ Primer Markovske verige:

Značilnosti Markovske verige

$\{X_k\}$ nereducirna, aperiodična, časovno homogena

- nereducirna ↓ se ne da minimizirati
- aperiodična ↓ aperiodična
- časovno homogena ↓ verjetnost prehoda med stanji se ne spreminja s časom

ERGODIČNA
M. VERIGA

Zanjo velja:

$$\Pi_j = \lim_{k\to\infty} \Pi_j^{(k)} = \lim_{k\to\infty} [X_k = j] \qquad j = 0,1,2\ldots$$

↑ verjetnost, da je v neskončnem času vrednost naklj. sprem. j.

$\equiv \tilde{\pi} e = 1$ def.

$$\sum_i \Pi_j = 1$$ Vsota verjetnosti vseh stacionarnih stanj je 1.

$$\Pi_j = \sum_i \Pi_i P_{ij} \equiv \tilde{\pi} = \tilde{\pi} P$$

Poglejmo si stac. verjetnost za naš primer z dvigalom

$$\tilde{P} = (p_{ij}) = \begin{array}{cc} & \begin{array}{ccc} \text{NASLI.} & & 2 \end{array} \\ \text{TRENU} & \begin{bmatrix} 0 & 1/2 & 1/2 \\ 3/4 & 0 & 1/4 \\ 3/4 & 1/4 & 0 \end{bmatrix} \end{array}$$

$\frac{1}{2}\ \frac{5}{4}\ \frac{3}{8}$

$$\pi_0 = \pi_0 \cdot 0 + \pi_1 \cdot 3/4 + \pi_2 \cdot 3/4$$
$$\pi_1 = \pi_0\, 1/2 + \pi_1 \cdot 0 + \pi_2 \cdot 1/4$$
$$\pi_2 = \pi_0\, 1/2 + \pi_1 \cdot 1/4 + \pi_2\, 0$$
$$\pi_0 + \pi_1 + \pi_2 = 1$$

Rešimo sistem

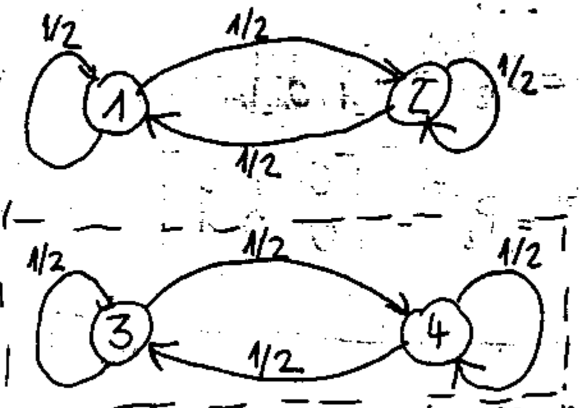
$$\boxed{\begin{aligned} \pi_0 &= \tfrac{3}{7} \\ \pi_1 &= \tfrac{2}{7} \\ \pi_2 &= \tfrac{2}{7} \end{aligned}}$$

Če bi se z dvigalom $k \to \infty$ -krat peljal z dvigalom, je verjetnost, da smo v pritličju $\frac{3}{7}$, ipd.

Markovska veriga se ne da minimizirati

$$\tilde{P} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

Diagram stanj:
(se da minimizirati)



Markovska veriga, ki se ne da minimizirati je tista pri kateri iz enega stanja lahko pridemo v drugega (v poljubnih korakih)

Markovska veriga, ki se ne da minimizirati, nima takih zaprtih sistemov

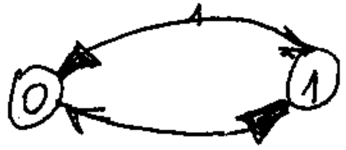
PERIODIČNA je tista, pri kateri se po $\mathbb{J}$ prehodih spet znajdemo v stanju, kjer smo začeli

$$\hat{P} = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \end{bmatrix}$$

Enaka matrika, ki se jo da minimizirati (le stanja so zamenjano)

$$\tilde{P}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

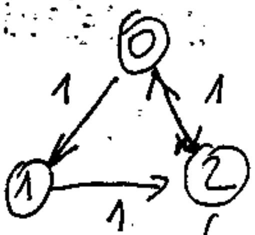
Primer periodične (matrike) markovske verige



Markovska veriga s periodo 1.

$$\tilde{P}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Še en primer periodične markovske verige

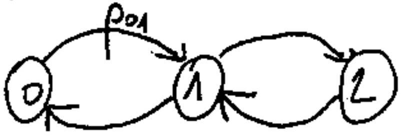


ZNAČILNOSTI M.V.

$$\tilde{P}_1 = \tilde{P}_1^{(3)} = \tilde{P}_1^{(5)} = P_1^{(k+1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \left(P_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right)$$

$$\tilde{P}_2 = P_2^{(4)} = P_2^{(7)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Petek, 6. 11. '98



↑ Zadnjič: markovske verige

MARKOVSKI PROCES ↘

MARKOVSKA VERIGA V ZVEZNEM ČASU

$\{x(t)\}$ $t_1 < t_2 < \ldots < t_k$ čas narašča

$$P[X(t_{k+1}) = j \mid X(t_1) = i_1, X(t_2) = i_2, \ldots X(t_k) = i_k] =$$

$$\boxed{\{x(t)\} = P[X(t_{k+1}) = j \mid X(t_k) = i_k]}$$

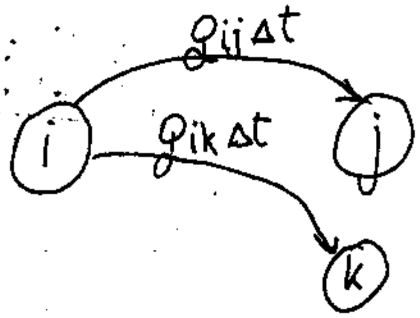
Čas od $t - t_{k-1}$ nas ne zanima. Pomembno je in vpliva v katerem stanju sem. Le da imema tu ZVEZEN ČAS.

DELO Z VERJETNOSTMI

<u>Ne</u> govorimo več o verjetnosti prehoda med stanjema i in j, pač pa o <u>verjetnosti prehoda med i in j v izredno kratkem času Δt</u>:

$$\boxed{P_{ij}(t, t+\Delta t) = \underset{\uparrow}{q_{ij}}\, \Delta t}$$

intenzivnost zapuščanja stanja i v stanje j



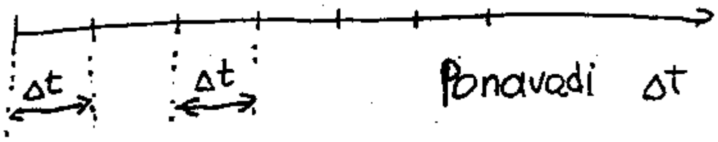
<u>Vsota vseh q-jev mora biti 1!</u>

$\sum_{j\neq i} q_{ij}$ celotna intenzivnost zapuščanja stanja

$\sum_{j\neq i} -q_{ij}$ celotna verjetnost da ne zapuščamo stanja

Čas, kolikor ga porabimo za stanje i je odvisen le od stanja oz. stoh. procesa, ki opisuje ta sistem.

τ_i - čas, ki ga sistem porabi v stanju i, preden naredimo prehod kamorkoli.



Ponavadi Δt limitiramo proti nič

čas, ki nas zanima:

$t = k\cdot\Delta t$ t - celotni čas (opazujemo posamezne rezine)

<u>verjetnost prehoda:</u>

1. $\sum_{j\neq i} q_{ij}\Delta t$, verjetnost da ostanemo v stanju i $1-\sum_{j\neq i} q_{ij}\Delta t$

čas prebivanja v stanju i:

vedno lahko govorimo o verjetnosti. To pa zato, ker je to stohastičen proces (primer kdaj se začnejo predavanja)

verjetnost, da je naš čas zadrževanja v i večji od t

$$P[\tau_i > t] = \lim_{\substack{k\to\infty \\ \Delta t\to 0}} \Big[1 - \sum_{i\neq j} g_{ij}\Delta t\Big]^k$$

$$\Downarrow$$

$$\Rightarrow \boxed{P[\tau_i > t] = e^{-g_i t}}$$

g_i - int. zapuščanja stanja

$$g_i = \sum_{i\neq j} g_{ij}$$

$$\boxed{P[\tau_i < t] = 1 - e^{-g_i t}}$$

običajna eksponentna distribucija

verjetnost, da smo v nekem času v določenem stanju

$$\pi_j(t) = P[X(t) = j]$$

$\pi_j(t)$ - verjetnost stanja v času t

$$\boxed{\pi_j(t+\Delta t) = \sum_{i\neq j} \pi_i(t) g_{ij}\Delta t + \pi_j(t)\Big[1 - \sum_{k\neq j} g_{jk}\Delta t\Big]}$$

verjetnost, da bomo v času $t+\Delta t$ v stanju j

=

← sprememba po času ($\Delta t \to 0$) odvod

$$\frac{d}{dt}\pi_j(t) = \sum_{i\neq j}\pi_i g_{ij} - \pi_j \sum_{k\neq j} g_{jk}$$

$$g_{ij} = -\sum_{k\neq j} g_{jk}$$

$$\boxed{\frac{d}{dt}\pi_j(t) = \sum_i \pi_i(t)\, g_{ij}}$$

↑ sprememba stanja po času

verjetnost, (če smo bili) v stanju i, intenzivnost prehajanja iz i v j (g_{ij}) krat Δt

\+ verjetnost da smo že v tem stanju krat (1 - intenzivnost prehajanja iz stanja j v stanje ℓ)

verjetnost, da ostanemo v stanju j

da smo

$$\tilde{\pi}(t) = (\pi_1(t), \pi_2(t), \dots)$$

$$\frac{d\,\tilde{\pi}(t)}{dt} = \left(\frac{d}{dt}\pi_1(t), \frac{d}{dt}\pi_2(t) \dots\right)$$

intenzivnost ostajanja

intenz zapuščanja

$$\tilde{Q} = (q_{ij}) = \begin{bmatrix} -\sum\limits_{j\neq 1} q_{1j} & q_{12} & q_{13} & \dots \\ q_{21} & -\sum\limits_{j\neq 2} q_{2j} & q_{23} & \dots \\ \vdots & \vdots & \vdots & \ddots \\ & & & -\sum\limits_{j\neq n} q_{nj} \end{bmatrix} \begin{matrix} \to \text{vsota} = 0 \\ \to \text{—||—} \\ \\ \to \text{vsota} = 0 \end{matrix}$$

vsota posame
vrstic je enaka

$$\boxed{\frac{d}{dt}\tilde{\pi}(t) = \tilde{\pi}(t)\tilde{Q}}$$

$\frac{d}{dt}\pi(t) = 0$ (za homogene markovske verige)

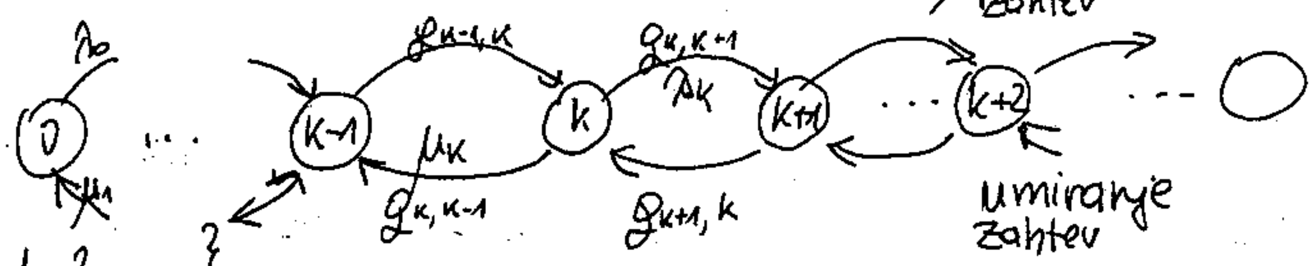
$\lim\limits_{t\to\infty} \tilde{\pi}(t) = \tilde{\pi}$ (stacionarno stanje)

$\tilde{Q}$ matrika intenzivnosti prehodov (zvezen čas)

$\tilde{P}$ vejetnostne matrike (diskreten čas)

ROJSTNO – SMRTNI PROCES

prehajamo lahko samo med sosednimi stanji



$\{0, 1, 2, \dots\}$

Število zahtev v sistemu se veča (porojevanje) ali manjša (umiranje zahtev). Tu se stvari bistveno poenostavljajo.

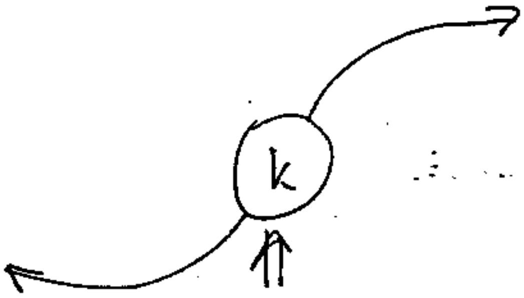
$\lambda_k = q_{k,k+1}$ porojevanje je enako intenzivnosti prihajanja zahtev

$\mu_k = q_{k,k-1}$ umiranje je enako intenzivnosti strežbe

$$Q = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 & \cdots \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & 0 & \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & 0 & \\ 0 & 0 & \mu_3 & -(\lambda_3+\mu_3) & \lambda_3 & \cdots \end{bmatrix}$$

$$\frac{d}{dt} P_k(t) = -(\lambda_k+\mu_k) P_k(t) + \lambda_{k-1} P_{k-1}(t) + \mu_{k+1} P_{k+1}(t) \qquad k \geq 1$$

Spremembe je odvisna od tega koliko se bo porodilo ali umrlo zahtev



odvisno od koliko zahtev se bo porodilo in koliko jih bo umrlo in če smo v stanju k+1, oz k-1

umiranje ↓ k

↓ porojevanje k

$$\frac{d}{dt}P_0(t) = -\underbrace{\lambda_0 P_0(t)}_{\substack{\text{verjetnost,}\\ \text{da si v}\\ \text{stanju 0}}} + \underbrace{\mu_1 P_1(t)}_{\substack{\text{verjetnost,}\\ \text{da iz}\\ \text{stanja 1 prideš}\\ \text{v stanje 0}}} \qquad *$$

LIMITNI PROCES PRI ROJSTNO-SMRTNEM PROCESU

$$\lim_{t\to\infty} P_k(t) = P_k = \pi_k$$ (stacionarno stanje)

(opazujemo dovolj dolgo) $\frac{dP_k}{dt} = 0$

vstavimo * !

$$0 = -(\lambda_k + \mu_k)\pi_k + \lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1}$$

$$0 = -\lambda_0 \pi_0 + \mu_1 \pi_1$$

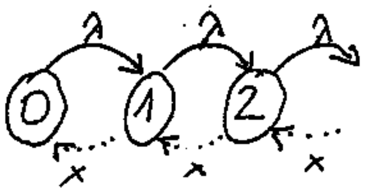
Primer:

$\lambda_k = \lambda > 0$ (intenzivnost porojevanja je enaka)

$\mu_k = 0$ (nimamo strežbe)

$P_0(0) = 0$ (začetno stanje)

$$\frac{dP_k(t)}{dt} = -\lambda P_k(t) + \lambda P_{k-1}(t)$$



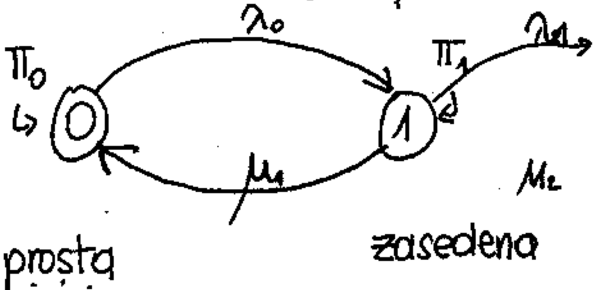
$$\frac{dP_0(t)}{dt} = -\lambda P_0(t)$$

Rešimo sistem enačb ⟹

$$\underbrace{P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}}_{\text{Poissonov proces}}$$

Primer:

Tel. centrala, pride klic in ko ga postrežemo, je centrala prosta



!

o izpit ?

$$0 = -\lambda_0\pi_0 + \mu_1\pi_1 \qquad (\lambda_0 = \lambda)$$
$$0 = -\mu_1\pi_1 + \lambda\pi_0 \qquad (\mu_1 = \mu)$$

enačbi sta dejansko enaki

Splošno:

$$0 = -(\underset{0}{\underset{\parallel}{\lambda_1}} + \mu_1)\pi_1 + \lambda_0\pi_0 + \underset{0}{\underset{\parallel}{\mu_2}}\pi_2$$

uravnotežna enačba

Dobimo:

$$\left.\begin{array}{l} \pi_1 = \frac{\lambda}{\mu}\pi_0 \\ \pi_1 + \pi_0 = 0 \end{array}\right\} \xrightarrow{\text{rezultat}} \underline{\underline{\pi_0 = \frac{1}{1+\frac{\lambda}{\mu}}}}$$

verjetnost, da smo v stanju 0

če sistem opazujemo dovolj dolgo.

MARKOVSKI PROCESI Z ENO ČAKALNO VRSTO (MM1) - TIPIČEN ROJSTNO-SMRTNI PROCES

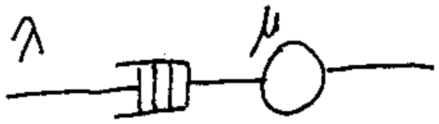
M/M/1

Poissonova porazdelitev na vhodu

eksp. porazd. strežbe (brez pom.)

1 strežnik

(brez pomnenja)



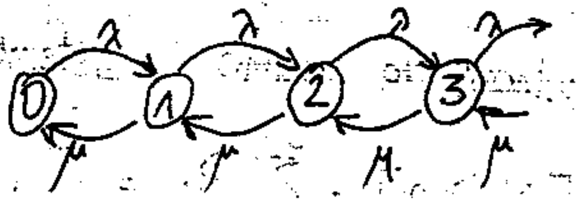
$N(t)$

$$\boxed{\begin{array}{l} t\to\infty \\ \frac{d}{dt}P_k(t) = 0 \end{array}}$$

Ne moreta se zgoditi 2 zahtevi hkrati (ker nimamo pomnenja).

$\frac{d}{dt}P_k(t) =$ $k \geq 1$ $P_k = \lim_{t\to\infty} P_k(t)$

$\frac{d}{dt}P_0(t) =$



$\lambda_k = \lambda$

$\mu_k = \mu$

uravnotežne enačbe:

$(\lambda+\mu)P_k = \lambda P_{k-1} + \mu P_{k+1}$ $k \geq 1$

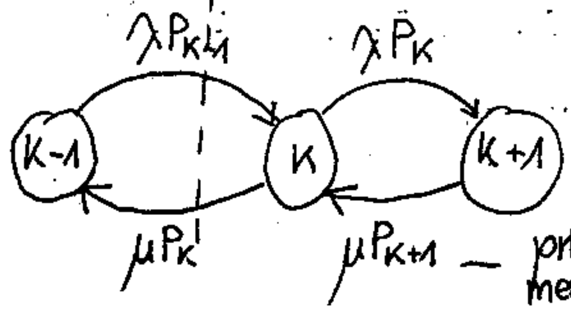
$\mu P_1 = \lambda P_0$

ρ - faktor uporabnosti

$\sum_k P_k = 1$ $P_0 = 1 - \frac{\lambda}{\mu} = 1 - \rho$ $\rho = \frac{\lambda}{\mu}$

$$\boxed{P_k = (1-\rho)\rho^k \quad \text{verjetnost, da je v sistemu } k\text{-zahtev}}$$

V splošnem:



$$(\lambda+\mu)P_k = \lambda P_{k-1} + \mu P_{k+1}$$ ← ravnotežje med posameznimi koeficienti

število prehodov iz ene v drugo smer mora biti enako:

$$\lambda P_k = \mu P_{k+1}$$

$$P_k = \frac{\lambda}{\mu} P_{k-1} = \left(\frac{\lambda}{\mu}\right)^2 P_{k-2} = \rho^k P_0 \qquad \rho = \frac{\lambda}{\mu}$$

$$P_0 = 1-\rho$$

$$P_k = (1-\rho)\rho^k$$

OSNOVNI POKAZATELJI SISTEMA

Verjetnost, da imamo v sistemu n ali več zahtev

$$P[N \geq n] = \sum_{k=n}^{\infty} P_k = \sum_{k=n}^{\infty}(1-\rho)\rho^k = (1-\rho)\sum_{k=n}^{\infty}\rho^k =$$

$$\boxed{P[N \geq n] = \underline{\underline{\rho^n}}}$$

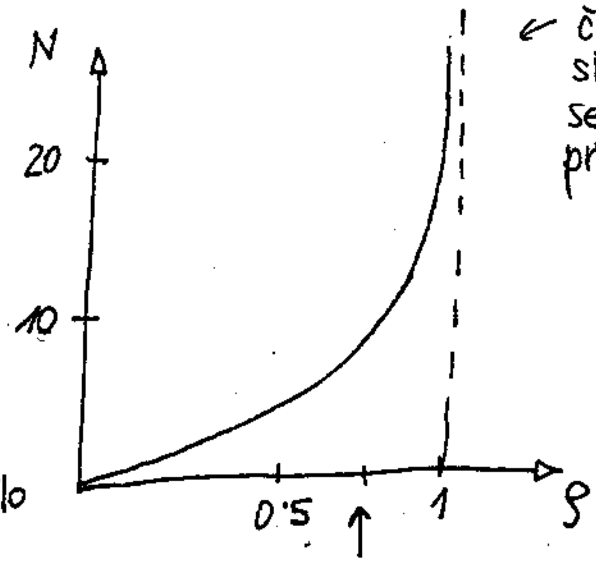
verjetnost, da imamo v sistemu n ali več zahtev

<u>Povprečno število zahtev v sistemu:</u>

$$N = \sum_{k=0}^{\infty} k P_k = \sum_{k=0}^{\infty} k(1-\rho)\rho^k = \ldots$$

$$\boxed{N = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}}$$

povprečno število zahtev v sistemu



← čakalne vrste gredo v ∞ sistem je v nasičenju. Zgodi se zaradi stohastičnega procesa prihajanja zahtev

75% je meja normalnega delovanja sistema, sicer so čakalni časi preveliki

Littlovo pravilo

$$\boxed{N=\lambda T}$$

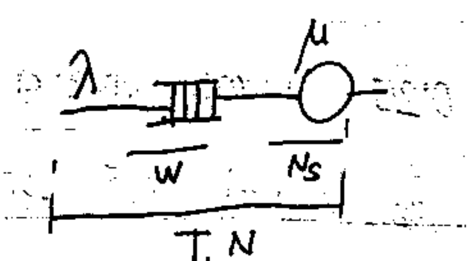
$$T=\frac{N}{\lambda}=\frac{\rho}{\lambda(1-\rho)}=$$

$$=\frac{\lambda}{\lambda(\mu-\lambda)}=\frac{1}{\mu-\lambda}$$ Povprečen čas zadrževanja v sistemu

$$\boxed{T=\frac{1}{\mu-\lambda}}$$

N_s - število zahtev v strežniku

$$N_s=\frac{\lambda}{\mu}=\rho=1-P_0$$



Povprečen čas v čakalni vrsti W:

povpr. čas strežbe $\boxed{\overline{x}=\frac{1}{\mu}}$

$$W=T-\frac{1}{\mu}=\frac{\rho}{\mu-\lambda}$$

$$T_s=\frac{N_s}{\lambda}=\frac{\frac{\lambda}{\mu}}{\lambda}=\frac{1}{\mu}$$ povp. čas zadrževanja v strežniku

$$N_q=\lambda W=\frac{\rho^2}{1-\rho}$$ število zahtev v vrsti

$$N_q=N-N_s=\frac{\rho}{1-\rho}-\frac{\rho(1-\rho)}{1-\rho}=\frac{\rho^2}{1-\rho}$$ -||-

Primer: Imamo tel. govorilnico in uporabniki prihajajo na 18 min.

$\lambda = \frac{1}{18}$ // $\frac{1}{18}$ uporabnikov/minuto
Poissonova verjetnost porazdelitve
FIFO vrsta, strežba

Čas strežbe je povprečna dolžina pogovora, ki je kar $\overline{x} = 7$ min

$\mu = \frac{1}{\overline{x}} = \frac{1}{7}$ uporabnikov/minuto.

Model M/M/1 uporabimo v tem primeru, ko imamo eksponentno funkcijo in poissonovo porazdelitev

⊛ $P_0 = 1 - \rho \qquad \rho = \frac{\lambda}{\mu} = \frac{\frac{1}{18}}{\frac{1}{7}} = \frac{7}{18}$ verjetnost, da bo sistem zaseden

⊛ Povprečna dolžina čakalne vrste

$N_Q = \frac{\rho^2}{1-\rho} = \frac{49}{198} \approx 0'25$ (skoraj prazna vrsta)

⊛ Kakšno je verjetnost, da je v vrsti več kot 5 ljudi v vrsti?

$P[N \geq 5] = \rho^5 \doteq 0{,}009$

Primer: Kdaj damo novo govorilnico? Recimo takrat, kadar uporabniki čakajo v vrsti več kot 7 minut.

$T \geq 7$ min → postavimo novo govorilnico
$\lambda' - ?$ – kakšno bo št. prihodov, da bo to potrebno?

V izPIT
o enačbe M/M/1
moraš znati

$\frac{\rho}{\mu-\lambda} = \frac{\frac{\lambda}{\mu}}{\mu-\lambda} \; ; \; \frac{\frac{\lambda'}{\mu}}{\mu-\lambda'} > 7$

$\frac{7\lambda'}{\frac{1}{7}-\lambda'} > 7 \; , \; 7\lambda' > 1 - 7\lambda'$

$14\lambda' \geq 1$

$\underline{\underline{\lambda' \geq \frac{1}{14}}}$ uporabniki prihajajo na 14 minut ali manj.

Primer

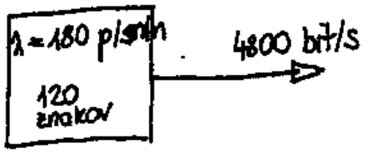
⊛ Komunikacijska naprava s poissonovo porazd. na vhodu

$\lambda = 180$ paketov /min ~ 3 paketi /s

Izhod modema je 4800 bit/s

Povprečna dolžina paketa je 120 znakov, lahko pa se spreminja po eksp. porazdelitvi. 1 znak ima 8 bitov.

IZPIT!



Zanima nas število paketov, dolžina vrste, verjetnost za 6 ali več zahtev v vrsti, ipd.

⊛ Povprečen čas strežbe

$$\overline{x} = \frac{120 \times 8}{4800} = 0'2 \text{ s}$$

120 znakov po osem bitov delimo s hitrostjo prenosa po serijski liniji

⊛ Faktor uporabnosti

$$\rho = \frac{\lambda}{\mu} = \lambda \overline{x} = 3 \text{ paketi/s} \cdot 0'2 \text{ s} = \underline{\underline{0'6}}$$

verjetnost zasedenosti linije je 60 %

⊛ Čas zadrževanja v sistemu

$$T = \frac{1}{\mu - \lambda} = \frac{1}{5-3} = \frac{1}{2} \text{ s}$$

$$\mu = \frac{1}{\overline{x}} = 5$$

⊛ Verjetnost, da je število zahtev v vrsti več ali enako 6

(Če bi imeli premalo bufferjev, bi paket lahko izgubili) Iščemo

! $\underline{\underline{N \geq 7}}$ ker je en paket v strežniku!

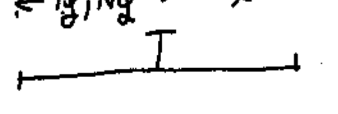
$$P[N \geq 7] = \rho^7 = 0{,}028 = \underline{\underline{2{,}8\,\%}}$$

⊛ Število zahtev v sistemu

Littlovo pravilo

$$N = \lambda T = 3 \cdot \frac{1}{2} = \underline{\underline{\frac{3}{2} \text{ paketa}}}$$

⊛ Čakalna vrsta



$$T_q = T - \overline{x} = \frac{1}{2}\text{s} - \frac{1}{5}\text{s} = \frac{3}{10} = \underline{0{,}3\text{s}}$$ povp. čas v vrsti

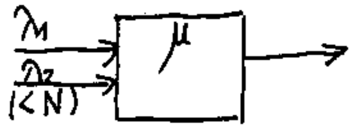
$$N_q = \lambda T_q = 3 \cdot 0{,}3 = 0{,}9 \text{ paketa}$$ povp. število zahtev v vrsti

Primer: Imamo komunikacijski sistem, ki prenaša govor in podatke

Če je sistem preveč zaseden, prenašamo samo govor, sicer govor in podatke. Analogija z banko, kjer čakajo tis ki imajo čas, ostali zapustijo banko, če je vrsta predolga (npr>5).

λ_1

λ_2



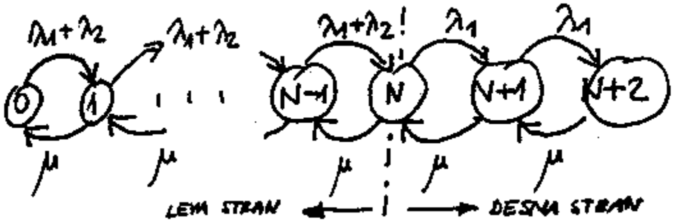
Sistem z različnim načinom porojevanja zahtev

Torej:

$\lambda_1 + \lambda_2 \quad k \leq N$

$\lambda_1 \quad k > N$

MARKOVSKA VERIGA ZA TA PRIMER:



Razlika z M/M/1 je v porojevanju.

⊛ Verjetnost stanja k

$$P_K = \begin{cases} \frac{\lambda_1+\lambda_2}{\mu} P_{K-1} \; ; & k \leq N \\ \frac{\lambda_1}{\mu} P_{K-1} \; ; & k > N \end{cases}$$

$\frac{\lambda_1+\lambda_2}{\mu}$

vstavimo v spl. enačba:

$$P_K = \left(\frac{\lambda_1+\lambda_2}{\mu}\right) P_{K-1} = \left(\frac{\lambda_1+\lambda_2}{\mu}\right)^2 P_{K-2} = \rho^k P_0 \quad ; \; k \leq N$$

$$P_K = \left(\frac{\lambda_1}{\mu}\right) P_{K-1} = \left(\frac{\lambda_1}{\mu}\right)^2 P_{K-2} = \rho_1^{K-N} P_N =$$

$$\rho_1 = \frac{\lambda_1}{\mu}$$

$$\underline{\underline{= \rho_1^{K-N} \rho^N P_0}}$$

Enačba je sestavljena iz dveh delov, če bi bila $\lambda_2 = 0$, bi dobili sistem M/M/1.

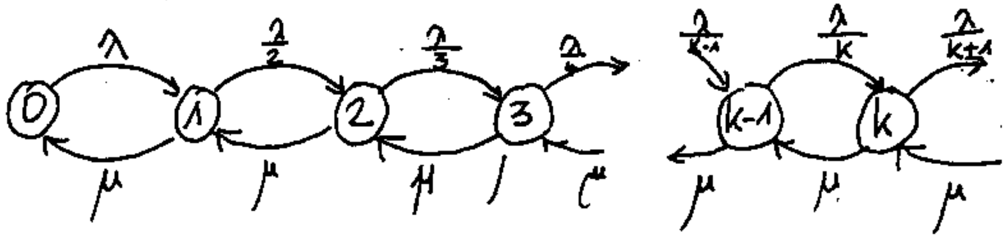
⊛ Reševanje takega sistema

$$P_0 \sum_{K=0}^{N-1} \rho^k + P_0 \sum_{K=N}^{\infty} \rho_1^{K-N} \rho^N = 1$$

$$\boxed{P_0 = \frac{1}{\frac{1-\rho^N}{1-\rho} + \frac{\rho^N}{1-\rho_1}}}$$

Primer: Imamo računalniški sistem "job processing". Intenzivnost porajevanja se zmanjšuje glede na število zahtev v sistemu (večje ko je št. zahtev v sistemu, manjša se intenzivnost novih zahtev) – omahljiv sistem. Takšen sistem praktično ne pride v nasičenje (da bi se čakalne vrste povečale v neskončnost)

$$\lambda_K = \frac{\lambda}{K+1} \qquad k=0,1,\dots$$



⊛ Verjetnosti – splošna enačba

$$P_K = \frac{\lambda}{\mu} P_{K-1}\,,$$ v našem primeru:
$$\boxed{P_K = \frac{\lambda_K}{\mu} P_{K-1}} = \frac{\lambda}{\overset{k}{\cancel{(k+1)}}\mu} P_{K-1} = \frac{\lambda^2}{k(k-1)\mu^2} P_{K-2}$$

IZPIT!

$$P_K = \frac{\rho^K}{K!} P_0\,,\quad \rho = \frac{\lambda}{\mu}$$

⊛ Vsota vseh verjetnosti je enaka 1

$$\sum_{k=0}^{\infty} P_0 \frac{\rho^k}{k!} = 1 \Rightarrow \boxed{P_0 = e^{-\rho}}$$

torej: $\boxed{P_K = \frac{\rho^k}{k!} P_0 = \frac{\rho^k}{k!} e^{-\rho}}$

$$\boxed{\bar{\lambda} = \sum_{k=0}^{\infty} \left(\frac{\lambda}{k+1}\right) P_K = \mu(1-e^{-\rho})}$$ povprečen λ je odvisen od strežbe.

⊛ Littlovo pravilo

$$T = \frac{N}{\bar{\lambda}} = \frac{\rho}{\mu(1-e^{-\rho})}$$

MODEL M/M/1/S

S - število zahtev v sistemu je omejeno

To je dejansko M/M/1 mdel, ki nima neskončne čakalne vrste. Če je v sistemu več kot S zahtev, nove zahteve padejo iz sistema in se ne postavljo v čakalno vrsto. Zato so čakalne vrste končne.

$$\boxed{P_K = \rho^K P_0} \quad \rho = \frac{\lambda}{\mu}$$

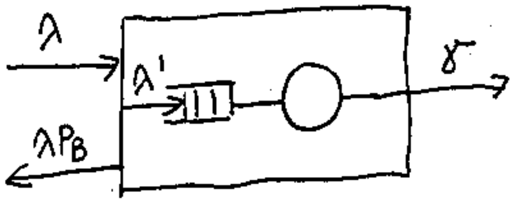
$K \leq S$

$K > S \rightarrow P_K = 0$

Uravnotežna enačba

$$\sum_{k=0}^{S} P_0 \rho^k = P_0 \sum_{k=0}^{S} \rho^k \qquad \boxed{P_0 = \frac{1-\rho}{1-\rho^{S+1}}}$$

$$\boxed{P_K = \frac{(1-\rho)\rho^K}{1-\rho^{S+1}}}$$



P_B - verjetnost, da je sistem blokiran

λP_B - intenzivnost zavračanja zahtev

$\lambda' = \gamma = \lambda(1 - P_B)$ Intenzivnost na izhodu

Intenzivnost strežbe * P_K je enaka γ.

$$\gamma = \sum_{K=1}^{S} \mu P_K = \mu \sum_{K=1}^{S} P_K = \mu(1 - P_0)$$

$$\lambda(1 - P_B) = \mu(1 - P_0)$$

Zanima nas verjetnost, da sistem ne sprejme zahteve.

$$P_S = \frac{(1-\rho)\rho^S}{1-\rho^{S+1}} \qquad \boxed{P_B = \frac{P_0+\rho-1}{\rho} = P_S}$$

Poseben primer:

$\rho=1$ - enako hitro strežemo kot prihajajo zahteve

$P_B = \frac{1}{S+1}$ P_B - verjetnost da pridemo na zaseden sistem. Bolj ko večamo število strežnikov, manjši je P_B

$$\boxed{N = \sum_{k=0}^{S} kP_k = \frac{\rho}{1-\rho} - \frac{\rho}{1-\rho}(S+1)P_S}$$

$\rho=1 \Rightarrow N = \frac{S}{2}$ Pri $\rho=1$ je povprečno število zahtev v sistemu $\frac{S}{2}$.

$$\boxed{N_S = 1 - P_0}$$

$$\boxed{N_q = N - N_S}$$

$\lambda' = \lambda(1-P_S)$

$T = \frac{N}{\lambda'}$

$W = \frac{N_q}{\lambda'}$

Pazimo, da je λ' dejanska (realna) obremenitev sistema

VEČJE ŠTEVILO STREŽNIKOV:

MODEL M/M/m

m - v sistemu imamo m enakovrednih strežnikov

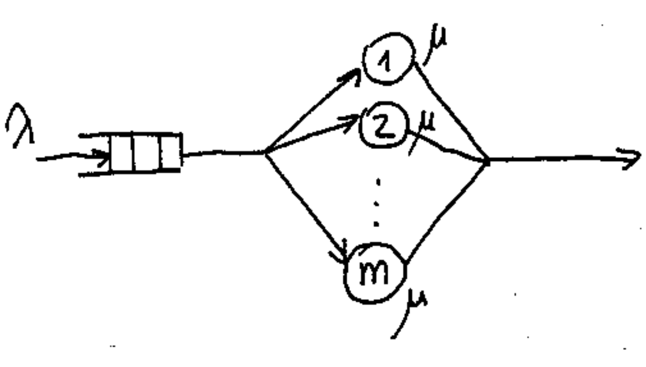
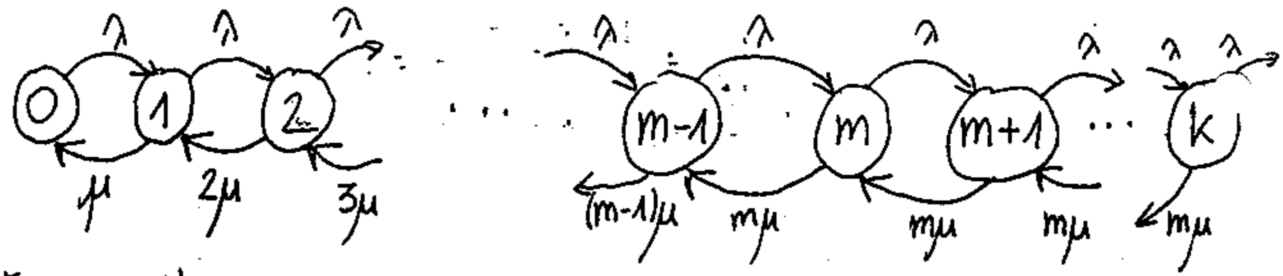


Diagram prehajanja stanja za M/M/m



Uravnotežne enačbe

verjetnost k-tega stanja: $P_k = \frac{\lambda}{k\mu} P_{k-1}$; $k \leq m$ $\boxed{\rho = \frac{\lambda}{m\mu}}$

$P_k = \frac{\lambda}{m\mu} P_{k-1}$; $k \geq m$

$k \leq m$: $\boxed{P_k = \frac{\lambda}{k\mu} P_{k-1} = \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} P_0}$ $k \geq m$: $\boxed{P_k = \frac{\lambda}{m\mu} P_{k-1} = \frac{\left(\frac{\lambda}{\mu}\right)^k}{m!\, m^{k-m}} P_0}$

$$P_k = \begin{cases} P_0 \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} ; & k \leq m \\ P_0 \frac{\left(\frac{\lambda}{\mu}\right)^k}{m!\, m^{k-m}} ; & k \geq m \end{cases}$$

$\sum P_k = 1$ $P_0 = \frac{1}{\sum_{k=0}^{m-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^m}{m!(1-\rho)}}$

Verjetnost, da moramo čakati (vsi strežniki so zasedeni)

!! izpiti podan P_d → enost. enačbe

$$\boxed{P_d = \sum_{k=m}^{\infty} P_k = \frac{P_0 \left(\frac{\lambda}{\mu}\right)^m}{m!\,(1-\rho)}}$$

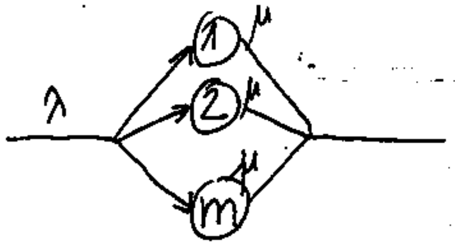
$N_q = \frac{\rho}{1-\rho} P_d$ $T = W + \frac{1}{\mu}$

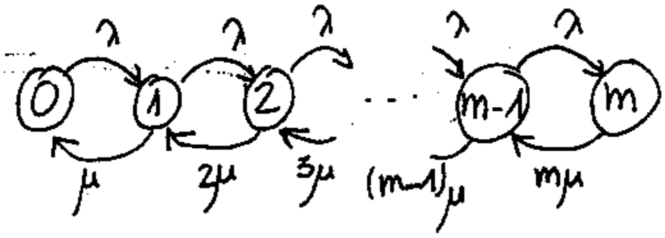
$W = \frac{N_q}{\lambda}$ $N = \lambda T$

Povprečen čas strežbe je enak, neglede na število strežnikov $\overline{X} = \frac{1}{\mu}$

MODEL M/M/m/m

Sistem brez čakalne vrste





$$P_k = \frac{\lambda}{k\mu} P_{k-1} = P_0 \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!}$$

$$P_0 = \frac{1}{\sum\limits_{k=0}^{m} \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k}$$

$$N = \sum_{k=0}^{m} k P_k = \frac{\lambda}{\mu}\left(1 - P_b\right) \qquad P_b = P_m$$

$$T = T_s = \frac{1}{\mu} \qquad (\text{ker je } \begin{matrix} W = 0) \\ N_q = 0 \end{matrix}$$

Primer je telefonska centrala.

PETRIJEVE MREŽE

Petrijeva mreža je sestavljena iz:

P - množica mest $\boxed{P = \{p_1, p_2, p_3, \dots, p_n\}}$ $n \geq 0$ } končna množica mest

T - množica prehodov $\boxed{T = \{t_1, t_2, t_3, \dots, t_m\}}$ $m \geq 0$ } končna množica prehodov

I - vhodna funkcija

O - izhodna funkcija

Oznaka za petrijevo mrežo je četvorka $C = (P, T, I, O)$

Množici P in T sta si tuji, torej $\boxed{P \cap T = \emptyset}$. Nimata nobenega skupnega elementa.

Vhodna funkcija I je funkcija, ki preslika množico prehodov v posplošeno množico ~~prehodov~~ mest. Prav tako je O.

$\boxed{I: T \to P^\infty}$

$\boxed{O: T \to P^\infty}$

lahko se isti element večkrat ponovi, v splošnem se lahko neskončnokrat ponovi

X^n - določen element množice se lahko ponovi n-krat

Primer petrijeve mreže

$C = (P, T, I, O)$

$P = \{p_1, p_2, p_3, p_4, p_5\}$ v petrijevi mreži imamo 5 mest

$T = \{t_1, t_2, t_3\}$ imamo tri prehode

Vhodne in izhodne funkcije nam opisujejo:

$I(t_1) = \{p_1\}$ Na vhodu t_1 je ~~vhodi~~ Za posamezno preh. imamo vh. mesta.

$I(t_2) = \{p_2, p_3, p_5\}$ Vhodna mesta za t_2 so p_2, p_3 in p_5

$I(t_3) = \{p_3, p_5\}$ ↖ ⊛

$O(t_1) = \{p_2, p_3, p_5\}$ Izhodna mesta za prehajanje t_1 so p_2, p_3 in p_5

$o(t_2) = \{p_5, p_5\}$

↓ mesto dvakrat nastopa na izhodu, primer posplošene množice

$O(t_3) = \{p_4\}$

4 ← s tem je naša petrijeva mreža opisana (torej $C = (P, T, I, O)$

⊛ Kako podam število vh. mest?

Število vhodnih mest označimo z znakom #

$\#(p_i, I(t_j))$	št. elementov p_i v posplošeni množici $I(t_j)$
$\#(p_2, I(t_2)) = 1$	v množici $I(t_2)$ je en sam element p_2
$\#(p_4, I(t_2)) = 0$	v množici $I(t_2)$ ni elementa p_4

To dejansko pomeni:
število mest p_i, ki vstopajo v prehod t_j.

⊛ Kako podam število izh. mest?

Podobno:

$\#(p_i, O(t_j))$	število elementov p_i v posplošeni množici $O(t_j)$.
$\#(p_5, O(t_2)) = 2$	število mest p_5 pri izh. mestu t_2

⊛ Kdaj je mesto vhodno?

Mesto p_i je vhodno mesto za prehajanje t_j, če velja

$p_i \cdot t_j \qquad p_i \in I(t_j)$

$p_1 \in I(t_1)$ - torej je p_1 vhodno mesto za prehajanje t_2

⊛ Kdaj je mesto izhodno?   Podobno

p_i je izhodno mesto, če je element funkcije $O(t_j)$.

$p_i \in O(t_j)$

⊛

Definicija: Vhodno funkcijo in izhodno funkcijo razširimo

$I: P \to T^\infty$ zamenjali smo mesta in prehoda.
$O: P \to T^\infty$

Zdaj bomo govorili o vhodnih (izhodnih) prehodih za mesta. Torej gledamo s stališča prehoda.

$$\boxed{\#(t_j, I(p_i)) = \#(p_i, O(t_j))}$$

V obe smeri je enako število prehodov

število prehodov, ki so vhodni mesto p_i je enako...

Enako velja:

$$\boxed{\#(t_j, O(p_i)) = \#(p_i, I(t_j))}$$

Nadaljujemo primer:

⊛ Število vhodnih prehodov za mesto p_1, p_2, p_3, p_4 in p_5.

$I(p_1) = \emptyset$

Za p_2: $I(p_2) = \{t_1\}$ Glej prejšnjo stran ⊛
p_3: $I(p_3) = \{t_1\}$
p_4: $I(p_4) = \{t_3\}$
p_5: $I(p_5) = \{t_1, t_2, t_2\}$ t_2 se dvakrat pojavi!

Izhodno funkcijo smo dejansko obrnili naokrog.

⊛ Podobno za izhode:

$O(p_1) = \{t_1\}$
$O(p_2) = \{t_2\}$
$O(p_3) = \{t_2, t_3\}$
$O(p_4) = \emptyset$
$O(p_5) = \{t_2, t_3\}$

Iz petrijeve mreže lahko narišemo petrijev graf.

PETRIJEV GRAF

Elementi petrijevega grafa so mesta, prehodi in usmerjene povezave, ki ustrezajo vhodnim in izhodnim funkcijam petrijeve mreže
Za mesta uporabimo krožec, za prehode uporabimo črte, za funkcije pa bomo risali puščice

Definicija

Petrijev graf G je bipartitni usmerjen graf $G = (V, A)$
V - množica vozlišč $\{v_1, v_2, \ldots v_s\}$
A - posplošena množica povezav $A = \{a_1, a_2, \ldots a_r\}$

$a_j = (a_j, v_k^t) \in V$

$V = P \cup T$ množica V je unija množice mest in množice prehodov

$a_i \in A$ Za vsako povezavo velja $a_i = (v_j, v_k)$ Za vsako izmed vozlišč velja

$> \quad v_j \in P \wedge v_k \in T$

$v_j \in T \wedge v_k \in P$

Začne se v prehodih in konča v mestih ali pa se začne v mestih in konča v prehodih

Vedno imamo kombinacijo prehod, mesto oz. mesto, prehod, nikoli pa nimamo kombinacije mesto, mesto oz. prehod, prehod.

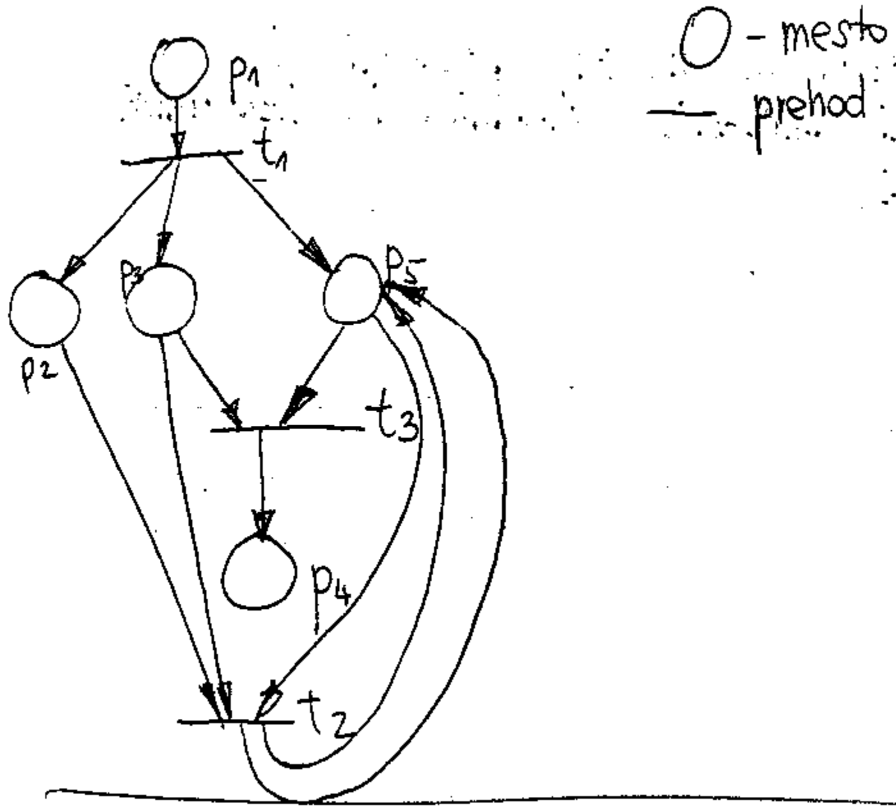
$A = P \cup T$

$p_i \in P$, $t \in T$

$$\boxed{\#((p_i, t_j), A) = \#(p_i, I(t_j))}$$

$$\boxed{\#((t_j, p_i), A) = \#(p_i, O(t_j))}$$

Narišemo petrijev graf:



f)

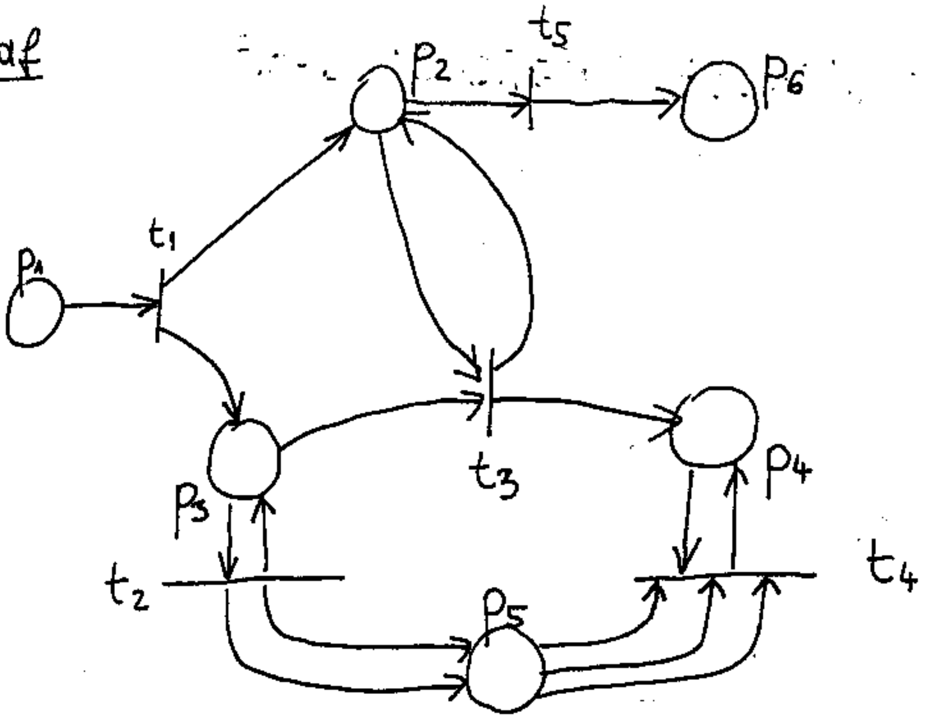
$C = (P, T, I, O)$

$P = \{p_1, p_2, p_3, p_4, p_5, p_6\}$

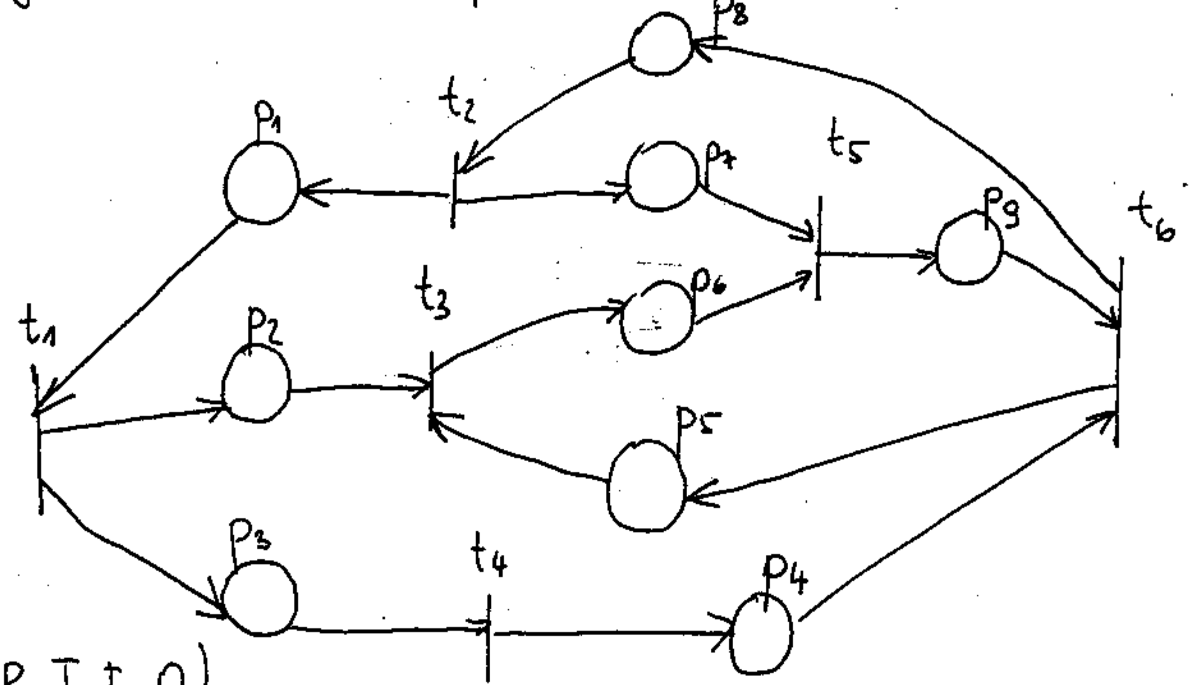
$T = \{t_1, t_2, t_3, t_4, t_5\}$

$I(t_1) = \{p_1\}$	$O(t_1) = \{p_2, p_3\}$
$I(t_2) = \{p_3\}$	$O(t_2) = \{p_3, p_5, p_5\}$
$I(t_3) = \{p_2, p_3\}$	$O(t_3) = \{p_2, p_4\}$
$I(t_4) = \{p_4, p_5, p_5, p_5\}$	$O(t_4) = \{p_4\}$
$I(t_5) = \{p_2\}$	$O(t_5) = \{p_6\}$

petrijev graf



Zdaj gremo obratno. Iz petrijevega grafa



$C = (P, T, I, O)$

$I(t_1) = \{p_1\}$ $\quad$ $O(t_1) = \{p_2, p_3\}$

$I(t_2) = \{p_8\}$ $\quad$ $O(t_2) = \{p_1, p_7\}$

$I(t_3) = \{p_2, p_5\}$ $\quad$ $O(t_3) = \{p_6\}$

$I(t_4) = \{p_3\}$ $\quad$ $O(t_4) = \{p_4\}$

$I(t_5) = \{p_6, p_7\}$ $\quad$ $O(t_5) = \{p_9\}$

$I(t_6) = \{p_4, p_9\}$ $\quad$ $O(t_6) = \{p_5, p_8\}$

$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_6, p_9\}$

$T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$

DUALEN IN INVERZEN PETRIJEV GRAF

Imamo petrijevo mrežo

$C = (P, T, I, O)$

$C_d = (T, P, I, O)$ Dualna petrijevo mreža
(obrnemo T in P)
obrnemo množico mest in množico prehodov

$C = (P, T, I, O)$

$T = \{t_1, t_2, t_3, t_4, t_5\}$ $P = \{p_1, p_2, p_3\}$

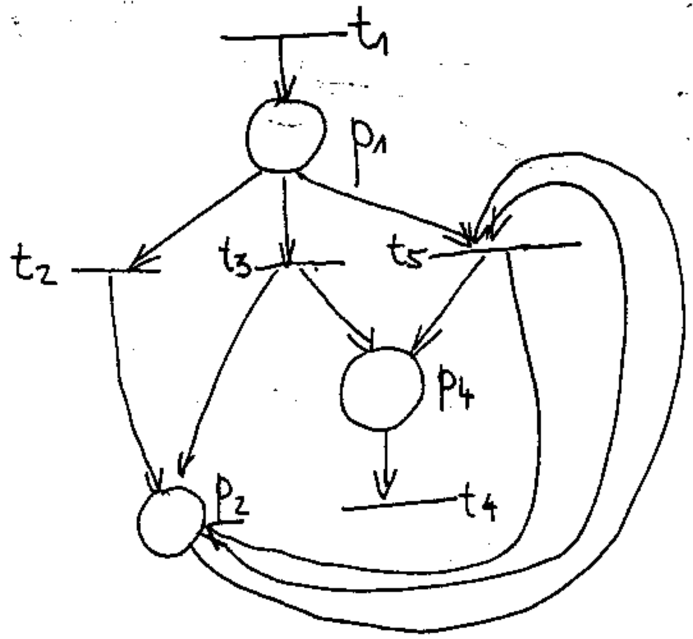
$I(t_1)$ $I(p_1) = \{t_1\}$ $O(p_1) = \{t_2, t_3, t_5\}$

$I(t_2)$ $I(p_2) = \{t_2, t_3, t_5\}$ $O(p_2) = \{t_5, t_5\}$

$I(t_3)$ $I(p_3) = \{t_3, t_5\}$ $O(p_3) = \{t_4\}$

$I(t_4)$

$I(t_5)$



$P = \{p_1, p_2, p_3, p_4, p_5\}$

$T = \{t_1, t_2, t_3\}$

INVERZEN
PETRIJEV GRAF

$O(t_1) = \{p_1\}$

$O(t_2) = \{p, p_3, p_5\}$

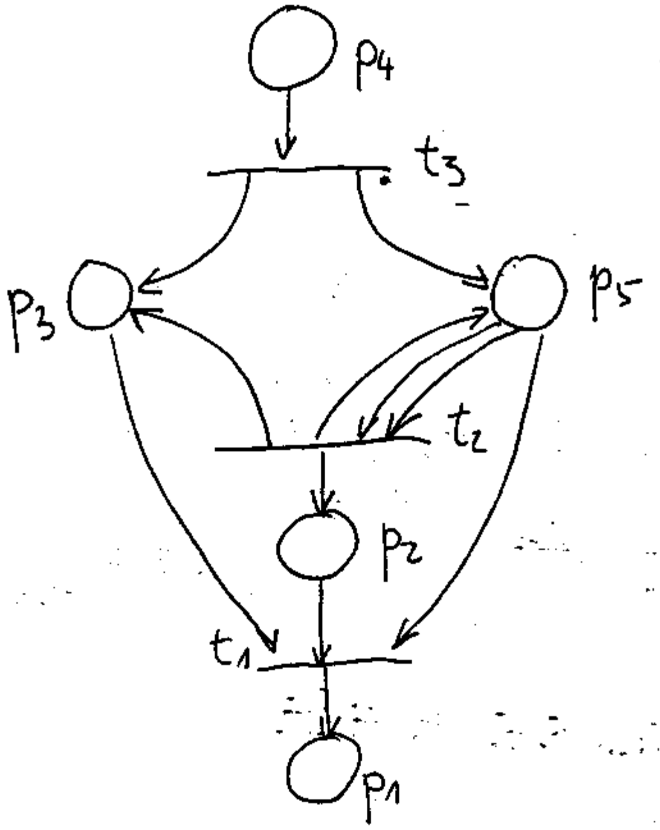
$O(t_3) = \{p_3, p_5\}$

$I(t_1) = \{p_2, p_3, p_5\}$

$I(t_2) = \{p_5, p_5$

$I(t_3) = \{p_4\}$

Smeri puščic se bodo spremenile.



OZNAČEVANJE PETRIJEVE MREŽE

Žetoni v sistemi so tisti ki se v sistemu spreminjajo, dejansko so to lahko spremenljivke. Število žetonov se lahko spreminja.

OZNAČITEV je

$C = (P, T, I, O)$ funkcija, ki mestom P priredi cela števila

$\sigma: P \rightarrow N$

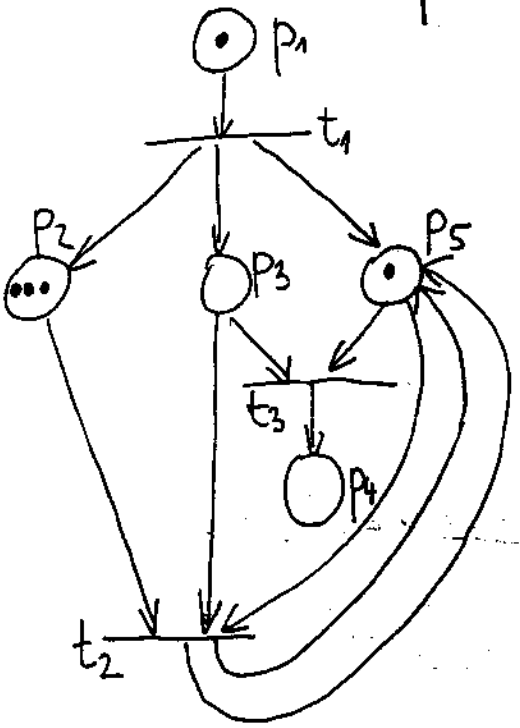
$\sigma = (o_1, o_2, ... o_n)$
oznacitveni vektor

pove koliko žetonov je ~~na~~ v posameznem mestu.

$M = (C, \sigma) = (P, T, I, O, \sigma)$

Primer: $\sigma = (1, 3, 0, 0, 1)$

Vsak žeton oznacimo s piko



Označen petrijev graf ima označena mesta

IZVAJANJE PETRIJEVE MREŽE

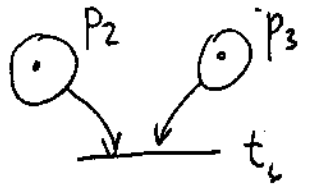
Izvajanje se začne, ko nek določen prehod vžge. Če bi t_1 vžgal bi vzel en žeton iz p_1 in ga damo v p_2, p_3 in p_5. Kaj pomeni jemanje in kaj dajanje žetonov. Če želimo, da prehod vžge mora biti petrijeva mreža označena, torej kadar [redacted] imamo podatke o žetonih

$t_j \in T$
$p_i \in P$

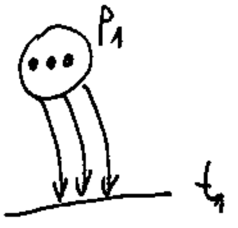
$\sigma(p_i) \geq \#(p_i, I(t_j))$

Prehajanje $t_j \in T$ v označeni pet. mreži M je izbrano oz. omogočeno če za vsa $p_i \in P$ velja naslednje: označitev mesta i je večje ali enako številu vseh mest za prehod j

Primer:



To mesto je izbrano, ker imamo v P_2 in P_3 vsaj en žeton



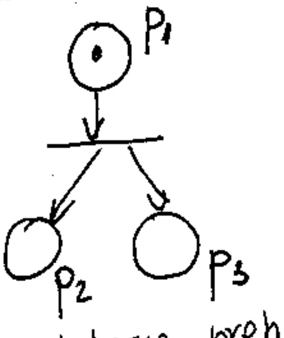
mesto t_1 je izbrano, ker imamo za vsak žeton eno puščico

Vsaka puščica zahteva svoj žeton na vhodu. Prehajanje lahko vžge, le kadar je izbrano (omogočeno). Ko prehajanje vžge, dobimo novo oznacitev.

NOVA OZNACITEV MESTA p_i

$$\boxed{\sigma'(p_i) = \sigma(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))}$$ vžig t_j

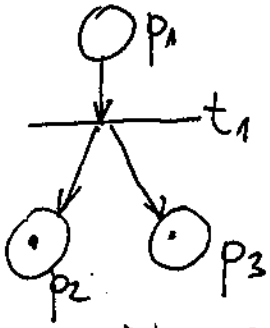
Primer



t_1 izbran prehod, ker je pogoj izpolnjen

t_1 izbrano

$\xrightarrow{\text{vžge}}$ ⟹

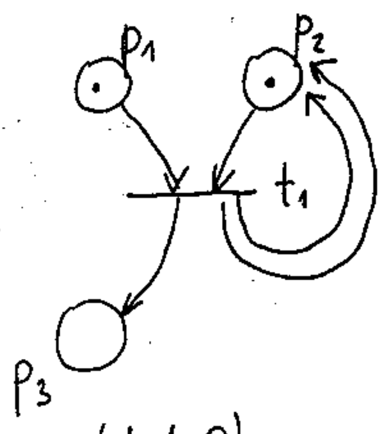


en žeton se je odvzel in dodal p_2 in p_3

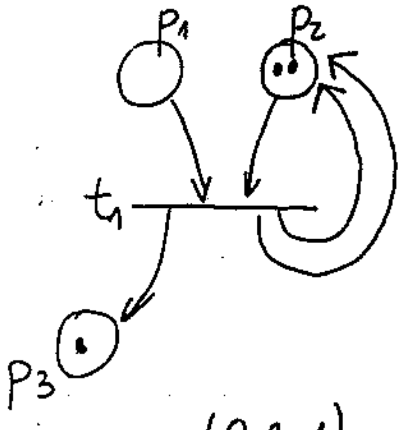
t_1 ni več izbrano

Izvajanje Petrijeve mreže se konča, ker nimamo več kaj izvajati, t_1 ni omogočen

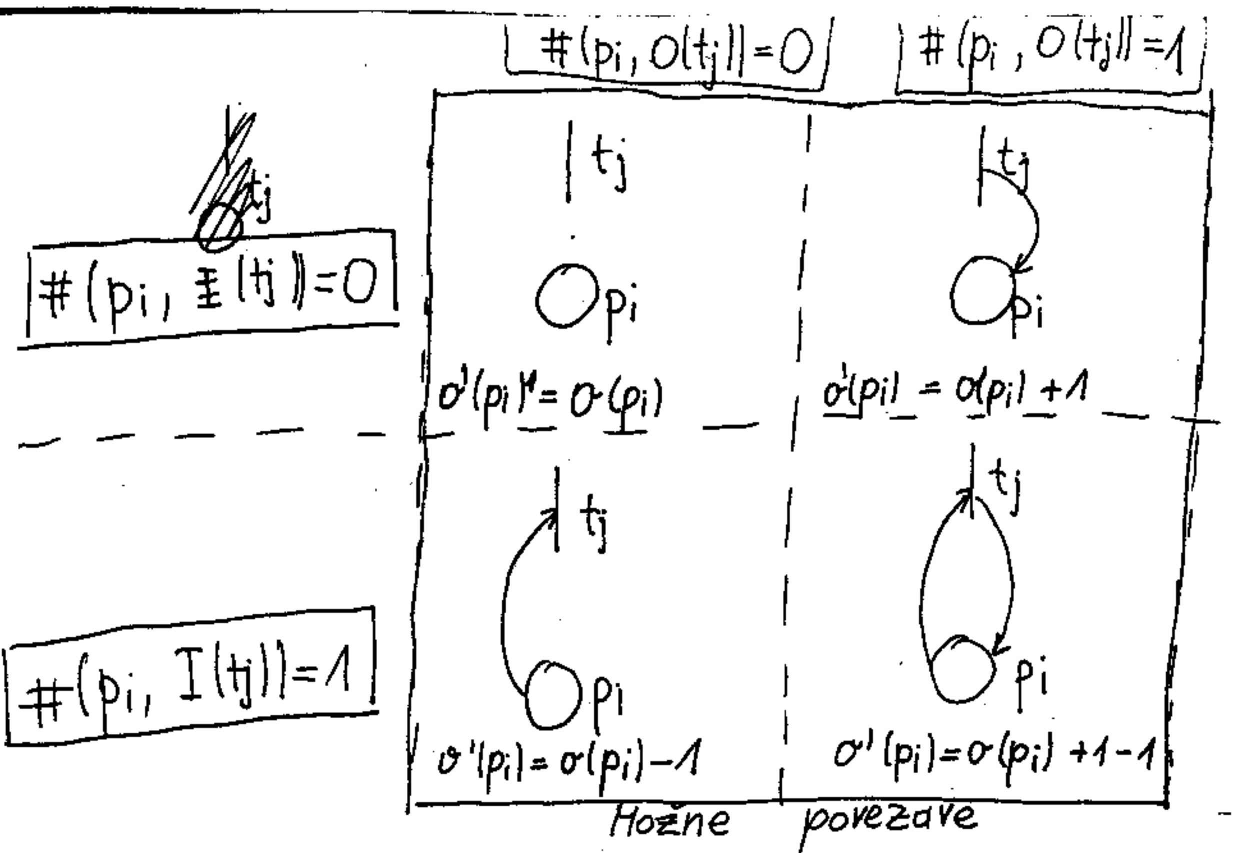
Primer



$\sigma=(1,1,0)$



$\sigma=(0,2,1)$



$\sigma'(p_i)$ – nova označitev

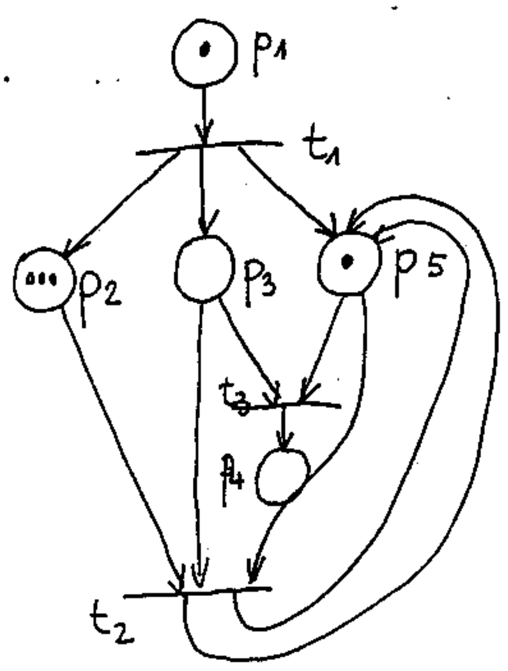
mesto p_i je izhodno

mesto p_i je izhodno in vhodno

Izvajanje petrijeve mreže lahko začnemo, ko je mesto izbrano.

Petek, 27. november 1998

Primer Petrijevega grafa

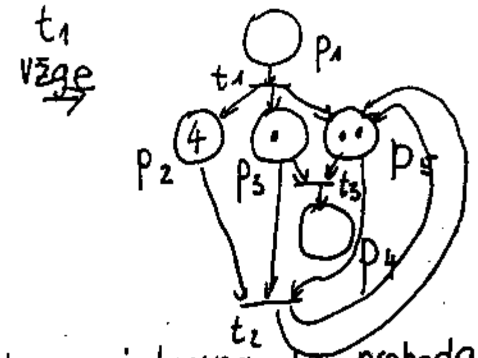


Če je v mestu p preveč žetonov, lahko le-te označimo s številom.

$\sigma = (1,3,0,0,1)$ ozn. vektor

Pogledati moramo, če je kakšno mesto izbrano. Prehod t_1 je izbran. t_2 : p_3 nima žetona in ne izpolnjuje pogoja, zato mesto ni izbrano. Podobno t_3 ni izbrano.

t_1 lahko vžge



Nova označitev

$\sigma' = (0,4,1,0,2)$

Mesto t_1: ni izbrano } ne izpolnjuje
mesto t_2: izbrano } izpolnjuje pogoj
mesto t_3: izbrano

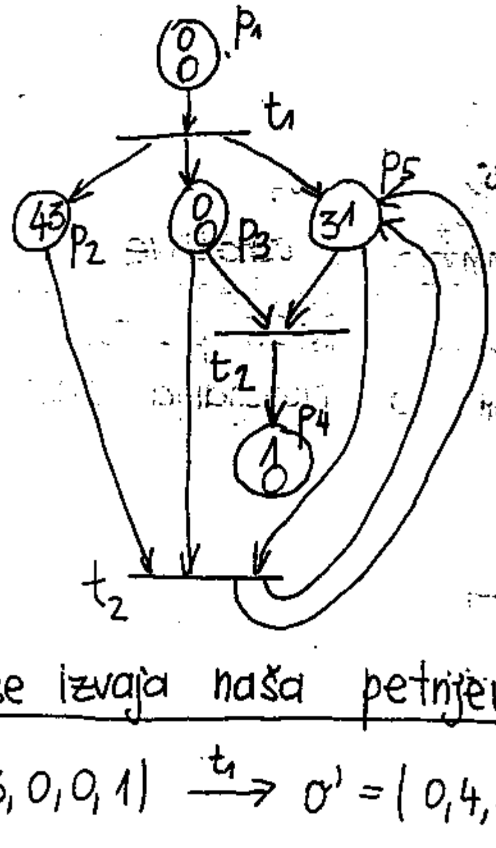
Imamo izbrana dva prehoda:

ni ne vhodno ne izhodno za prehod t_2

Če vžge $t_2 \rightarrow \sigma'' = (0,3,0,0,3)$

če vžge $t_3 \rightarrow \sigma'' = (0,4,0,1,1)$

Dva možna rezultata. Zanima nas, če se še kaj lahko izvajo. Poglejmo označitev po t_3 in t_2.

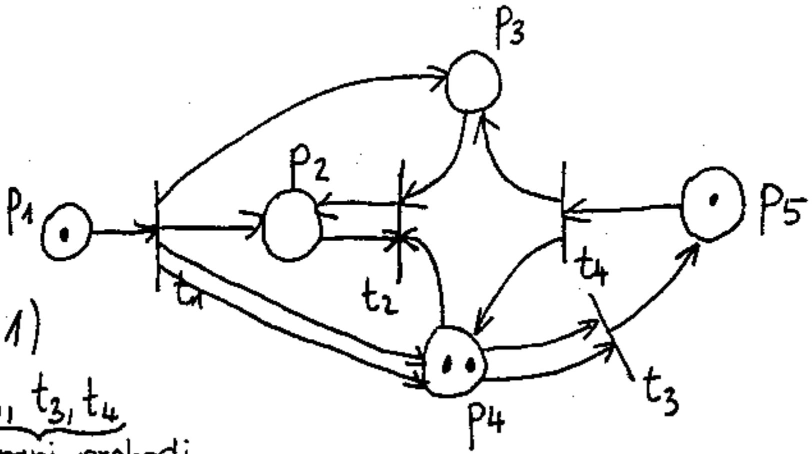


Vidimo, da nobeno mesto ni izbrano. Izvajanje Petrijeve mreže se tu konča.

Kako se izvaja naša petrijeva mreža:

$$\begin{tikzcd} \sigma = (1,3,0,0,1) \arrow[r, "t_1"] & \sigma' = (0,4,1,0,2) \arrow[r, "t_2"] \arrow[dr, "t_3"'] & \sigma'' = (0,3,0,0,3) \\ & & \sigma'' = (0,4,0,1,1) \end{tikzcd}$$

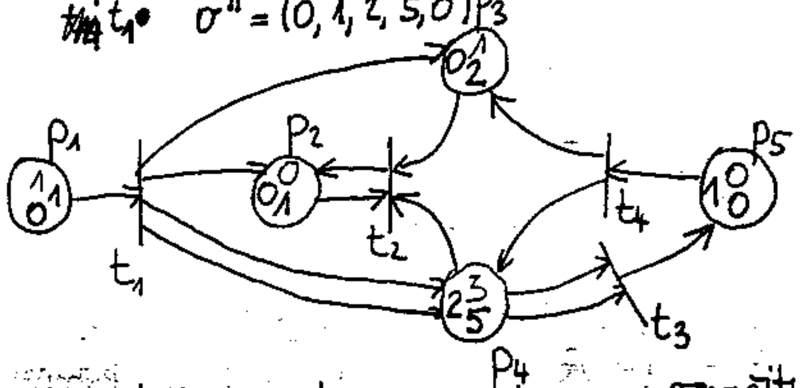
Primer



$\sigma = (1,0,0,2,1)$

Izbrana : $\underbrace{t_1, t_3, t_4}_{\text{Izbrani prehodi}}$

LEGENDA : t_1 • $\sigma = (1,0,0,2,1)$
t_4 • $\sigma' = (1,0,1,3,0)$
t_1• $\sigma'' = (0,1,2,5,0)$



Petrijev graf ostaja enak, spreminjajo se označitve.

$\sigma : N^m \times T \to N^m$ — ostajamo v istem prostoru
σ - funkcija prehajanja stanj

$\sigma(o, t_j)$ - Kakšna bo funkcija naslednjega stanja
imamo označitev in izbrano mesto

$\boxed{\sigma(o, t_j) = \sigma(o(t), t_j) = \sigma' = o(t+1)}$ - označitev v naslednjem koraku

Dobimo formalni zapis izvajanja petrijeve mreže.

Vžig petrijeve mreže se izvede v realnem času (preklopu).
t, t+1 sta logična časa, naslednji korak, ne pa nek absoluten čas.

PETRIJEVE MREŽE

SINHRONE
Vžig se izvede
- takoj, ko je izbrana
Pogojeno z neko
sinhronsko uro. (primer računalnika)

ASINHRONE
na vžig vpliva
zunanji dejavnik

Če je izbranih več prehodov, lahko določimo prioriteto posameznih prehodov. Druga možnost pa je, da se izvede več prehodov neodvisnih izbranih prehodov. Takrat uvedemo paralelizem. (primer: večprocesorski sistem)
Lahko pa mesta izberemo tudi naključno.

Označitev sekvenc ··· Sekvence aktivnih stanj

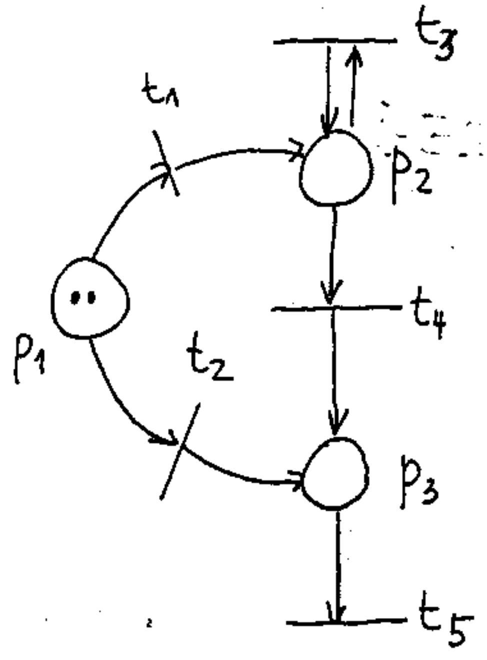
$O = \sigma^1 \sigma^2 \sigma^3 \ldots \sigma^r$ $T = t_{j_1}, t_{j_2}, \ldots$

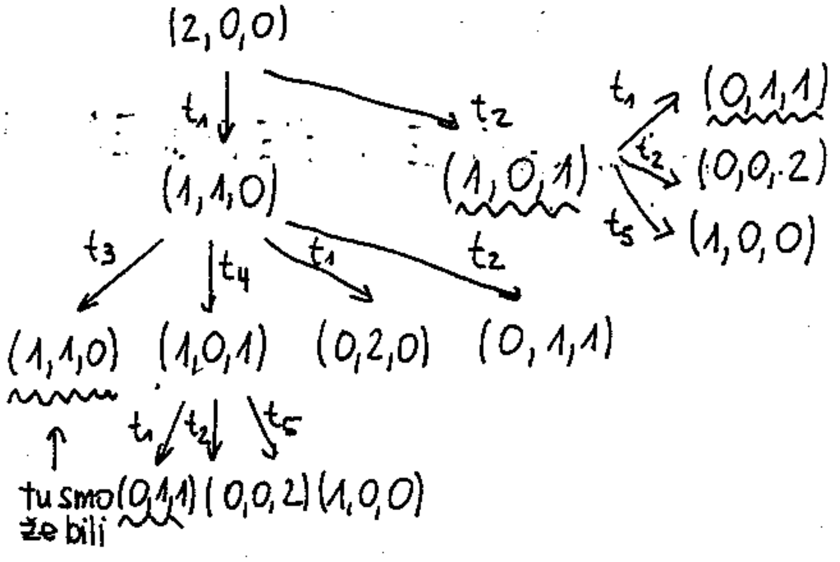
σ^1 – označitev vektor pri prvem prehodu

Dosegljivost petr. mreže so vsa stanja, ki jih lahko dosežemo iz začetne označitve vse označitve.

Primer računanja dosegljivosti: $C = (P, T, I, O)$

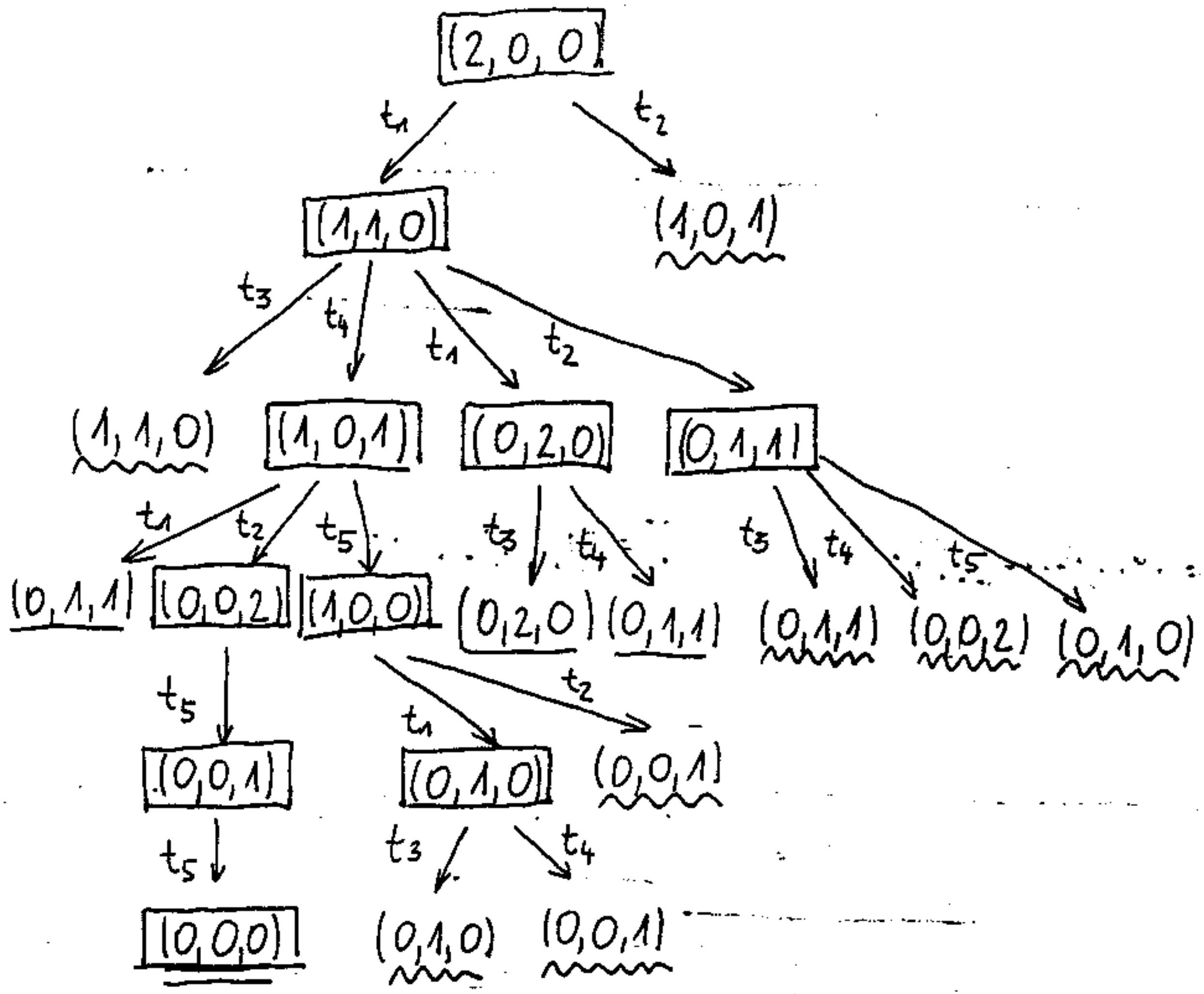
$R(C, \sigma)$ R-dosegljivost





→ NA NASLEDNJI STRANI

$\sigma = (2,0,0)$ začetni označitveni vektor
Iskali bomo vse možne označitve, do katere lahko pride petrijeva mreža



Ko pridemo do stanja, ki se nam ponovi, se ustavimo.
Dosegljivost so vsa stanja.

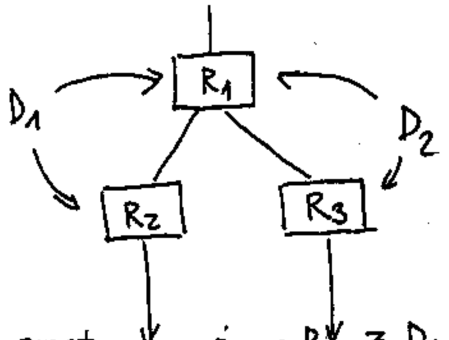
MODULIRANJE S PETRIJEVO MREŽO

Dogajanje, akcije, se morajo dogajati pod določenimi pogoji.
dogodki

Pogoji so tisti, ki nas vežejo. Če so pogoji izpolnjeni se dogodek lahko izvrši, kar pa nas že spominja na petrijevo mrežo

Imamo tri računalnike in dva operaterja (D) (R)

Pogoji bodo podani v pet. mreži



f - D_1 prost
g - D_2 prost
h - R_1 zaseden z D_1
i - R_1 z D_2

j - R_2 z D_1
k - R_3 z D_2
l - program je izvršen

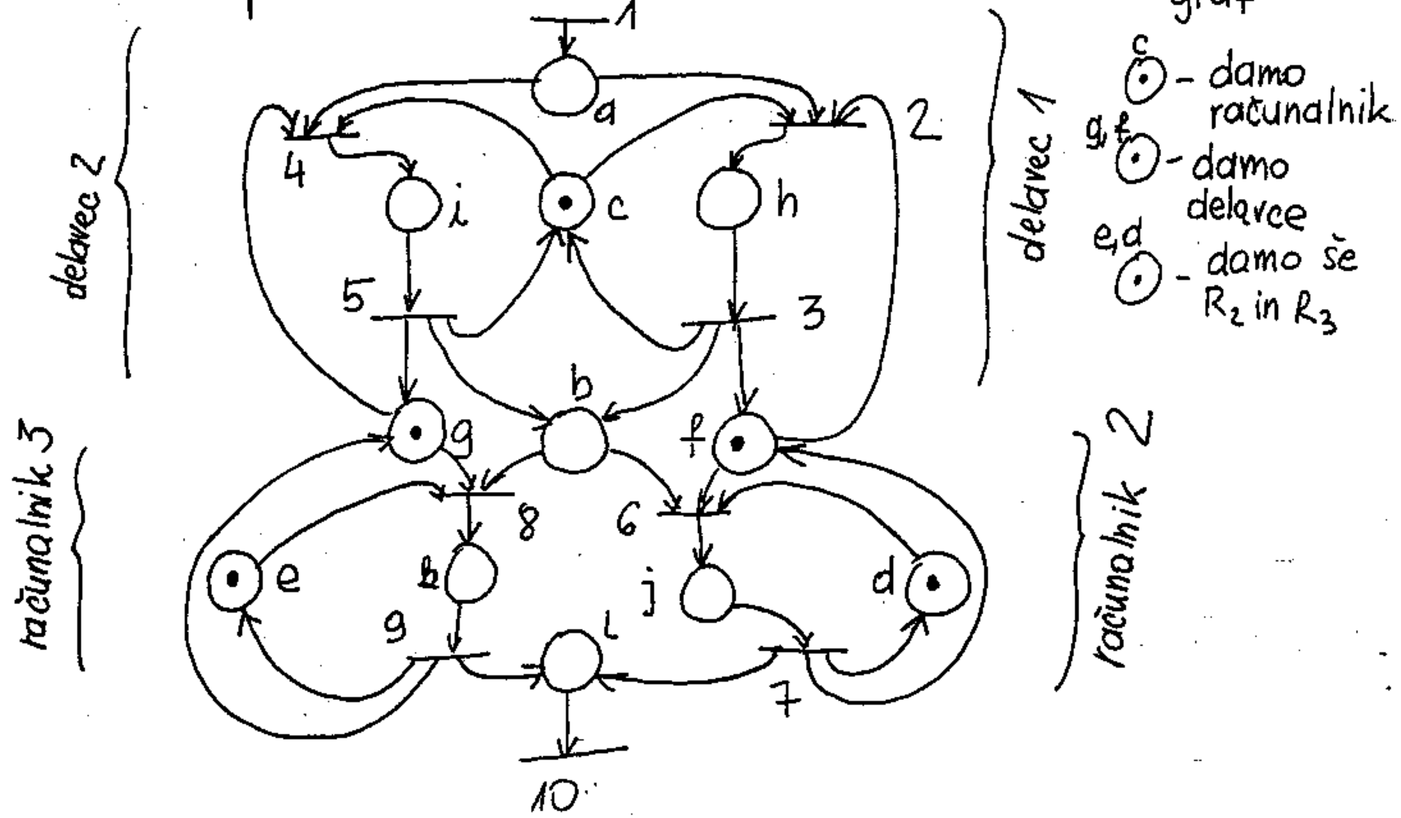
a - program čaka na R_1
b - program je bil izveden na R_1 in čaka da se izvede na R_2 ali R
c - računalnik 1 je prost
d - —||— 2 je prost
e - —||— 3 je prost

Našteli smo vse pogoje, ki se nam lahko izvedejo
Sedaj pa akcije:

1 - program dospe v obdelavo
2 - D_1 začne z delom na R_1
3 - D_1 konča z delom na R_1
4 - D_2 začne —II— R_1
5 - D_2 konča —II— R_1
6 - D_1 začne —II— R_2
7 - D_1 konča z delom na R_2
8 - D_2 začne —II— na R_3
9 - D_2 konča —II— na R_3
10 - END

Dogodek	predpogoj	po. pogoj (rezultat dogodka)
1	—	a
2	a, c, f	h
3	h	f, c, b
4	a, g, c	i
5	i	b, g, c
6	b, f, d	j
7	j	l, f, d
8	b, e, g	k
9	k	l, e, g
10	l	—

S tem smo opisali celotno mrežo in lahko narišemo petrijev graf



Ko imamo več žetonov (recimo dodamo delavca, računalnik) povečujemo stevilo resursov (računalnikov, delavcev), moduliranje pa se še vedno pravilno izvaja.

MODELIRANJE AVTOMATA S PETRIJEVO MREŽO

Avtomat opišemo s petorko $A = \{X, Y, Z, \delta, \lambda\}$

X... vhodna abeceda (pogoj za prehod v naslednje stanje)

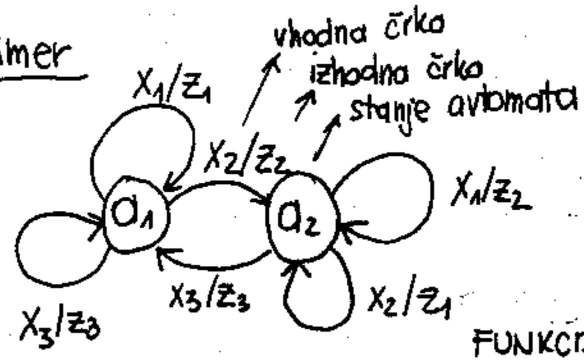
Y... množica stanj

Z... izhodna abeceda

δ... funkcija, ki pove kako se avtomat odziva v določenem stanje

λ... povezava med vhodno črko in notranjim stanjem in izhodno črko

Primer



Obstajata dva avtomata Meally-ev in Moorov avtomatov.

$X = \{x_1, x_2, x_3\}$

$Y = \{a_1, a_2\}$

$Z = \{z_1, z_2, z_3\}$

FUNKCIJA NASLEDNJA STANJA:	FUNKCIJE IZH. ČRKE
$\delta(a_1, x_1) \rightarrow a_1$	$\lambda(a_1, x_1) \Rightarrow z_1$
$\delta(a_1, x_2) \rightarrow a_2$	$\lambda(a_1, x_2) \Rightarrow z_2$
$\delta(a_1, x_3) \rightarrow a_1$	$\lambda(a_1, x_3) \rightarrow z_3$
$\delta(a_2, x_1) \rightarrow a_2$	$\lambda(a_2, x_1) \rightarrow z_2$
$\delta(a_2, x_2) \rightarrow a_2$	$\lambda(a_2, x_2) \rightarrow z_1$
$\delta(a_2, x_3) \rightarrow a_1$	$\lambda(a_2, x_3) \rightarrow z_3$

Kako deluje avtomat (MEALYEV)

VH. ČRKA		x_1	x_2	x_2	x_3	x_1	x_3	...
STANJE	a_1	a_1	a_2	a_2	a_1	a_1	a_1	...
IZH. ČRKA		z_1	z_2	z_1	z_3	z_1	z_3	...

← (izberemo naključno, da pokažemo delovanje)

Pravilnostna tabela kot opis avtomata(bolj pregleden zapis):

Informacija pa je enaka kot pri grafu.

	a_1	a_2
x_1	t_{11} a_1/z_1	t_{21} a_2/z_2
x_2	t_{12} a_2/z_2	t_{22} a_2/z_1
x_3	t_{13} a_1/z_3	t_{23} a_1/z_3

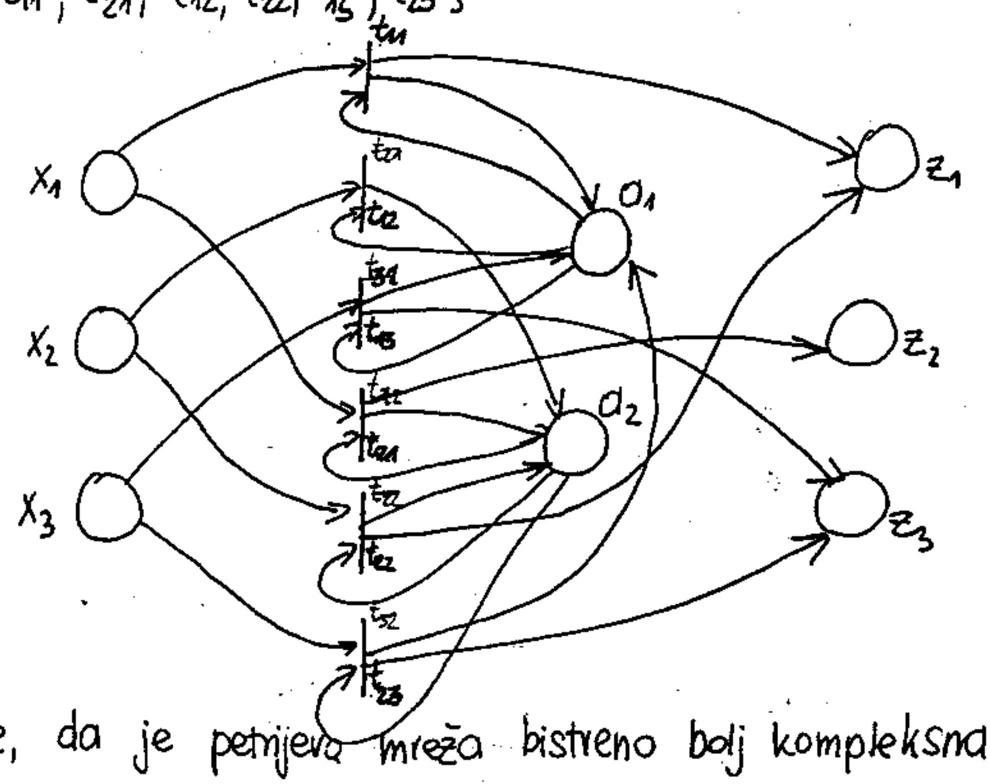
Tak avtomat želimo zapisati s petrijevo mrežo. Definirati moramo:

množica mest: $P = Y \cup X \cup Z$ imeli bomo 8 mest v petrijeni mreži

množica prehodov: $T = \{ t_{y,x} \mid y \in Y, x \in X \}$
Vsi pari notranjih stanj in izh. črke

pogoj za vh. množico: $I(t_{y,x}) = \{y, x\}$.
izh. množica: $O(t_{y,x}) = \{\delta(y,x), \lambda(y,x)\}$
za par x,y → če določen par vzge je par $\delta(y,x)$, $\lambda(y,x)$

$P = X \cup Y \cup Z = \{x_1, x_2, x_3, a_1, a_2, z_1, z_2, z_3\}$
$T = \{ t/_{a_1x_1}, t/_{a_2x_1}, t/_{a_1x_2}, t/_{a_2x_2}, t/_{a_1x_3}, t/_{a_2x_3} \}$
$T = \{ t_{11}, t_{21}, t_{12}, t_{22}, t_{13}, t_{23} \}$

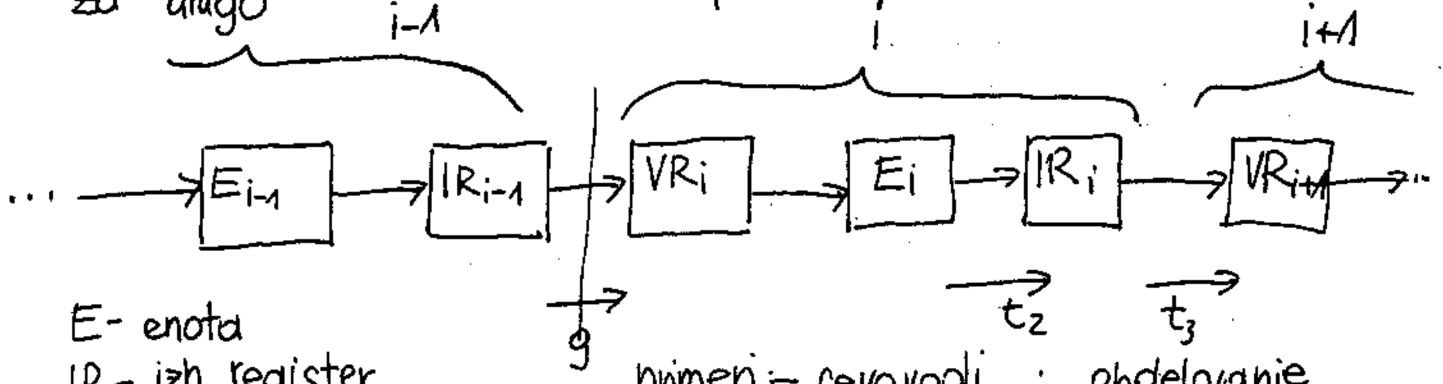


Izkaže se, da je petrijeva mreža bistveno bolj kompleksna, kot avtomat.

UPORABA PETRIJEVIH MREŽ

Primer

⊛ Sinhronizacija v računalniku, opravljanje nalog eno za drugo



E- enota
IR- izh. register

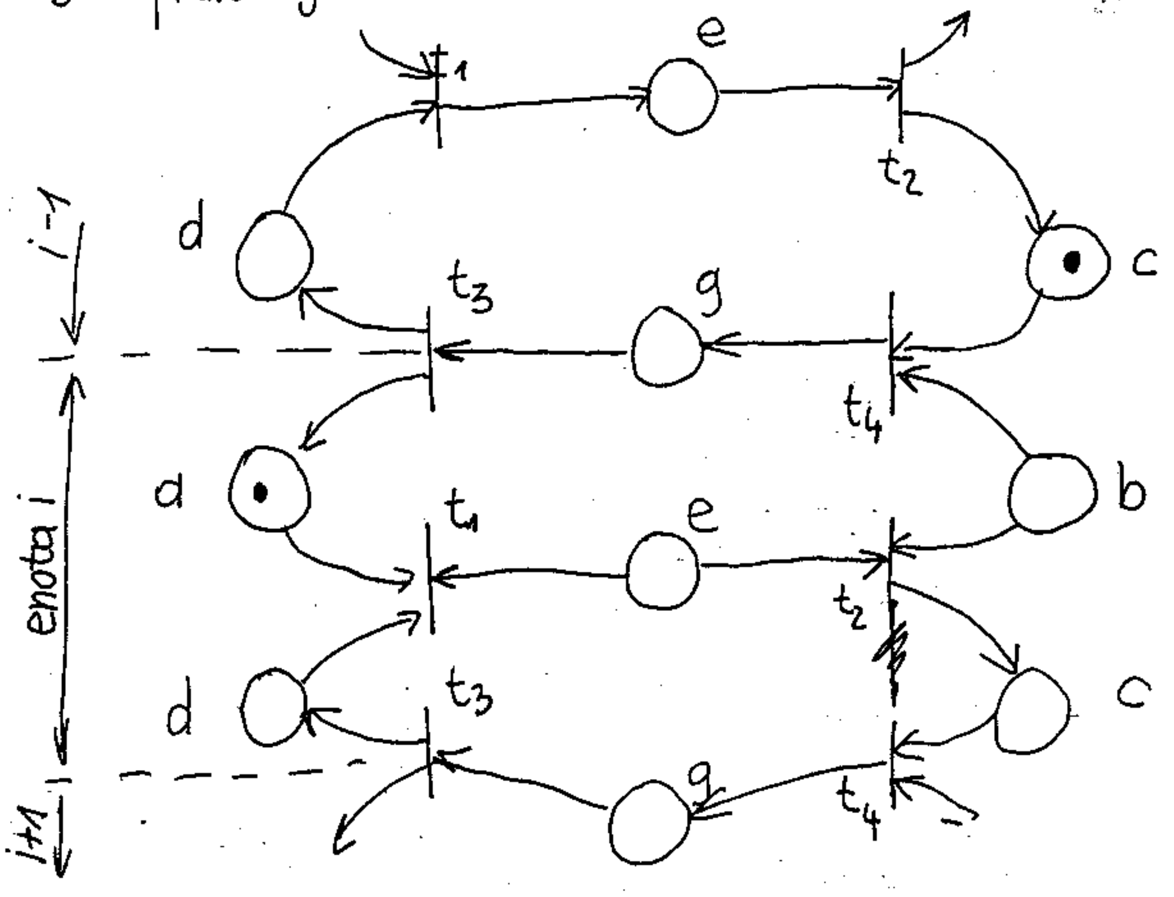
primeri – cevovodi : obdelovanje podatkov hitro enega za drugim. Ce je to rešeno hardversko, potrebujemo neko sinhronizacijo

mesta

a- vhodni register je zaseden
b)- vhodni register je prost
c - izh. register je zaseden
d - izh. register je prost
e - enota je zasedena
f - enota je prosta
g - prenos je možen

dogodki

t_1 - enota izvede akcijo (E_i)
t_2 - vpis v izhodni register i
t_3 - prenos IR_i v VR_{i+1}
t_4 - prenos vsebine v naslednjo enoto

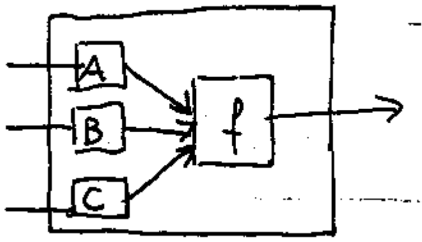


Na sliki nimamo f-enota je prosta. V začetku mora veljati b in d. V našem primeru f sploh ne potrebujemo.

vhodni izhodni
register prost.

Delovanje: Če je enota zasedena, morata biti vhodni in izhodni register prosta. Petrijeva mreža skrbi za sinhronizacijo, dogodki so vezani. Delovanje razberi iz petrijevega grafa.

⊗ Primer porazdeljenega procesiranja. Imamo procesni element z registri a, b, c. Izvajamo funkcijo nad temi podatki f(a, b, c)



DATA FLOW
poganjane enote s strani podatkov

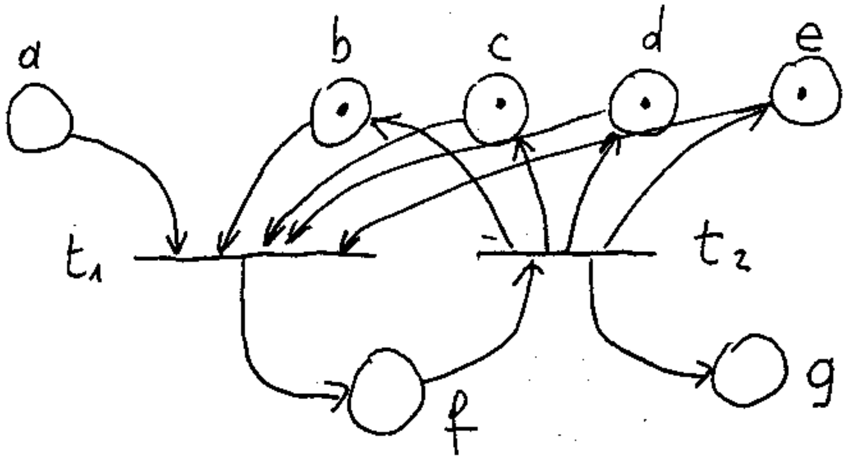
Podatki prožijo akcijo. Ko so podatki prisotni, se akcija izvrši. Ko se akcija izvrši, se registri spraznijo.

mesta:

a – aktiviranje procesnega elementa f
b – f prost
c – register A je prost
d – register B je prost
e – register C je prost
f – A, B, C, f zasedene
g – konec obdelave

dogodki, oz. prehodi

t_1 – izvajanje inštrukcije
t_2 – sprememba registrov po koncu delovanja

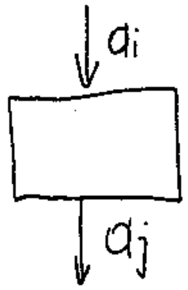


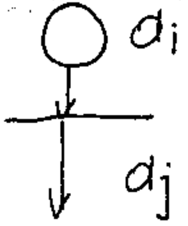
Na začetku se registri zaklenejo. Ponavadi imamo več ALE enot. Ne smemo spreminjati registrov za cikel, ko se izvaja operacija.

⊛ <u>Zapišimo program v obliki petrijeve mreže</u>

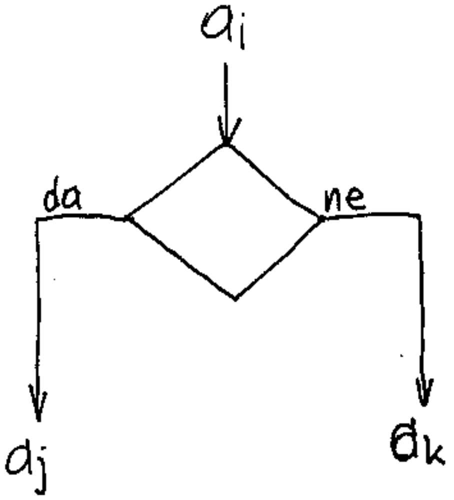
Programi v računalniku komunicirajo z množico ostalih programov (oz. aparaturno opremo). Komunikacijski algoritmi morajo obvezno biti <u>sinhronizirani</u>. Komunikacijski protokoli to stvar rešujejo.

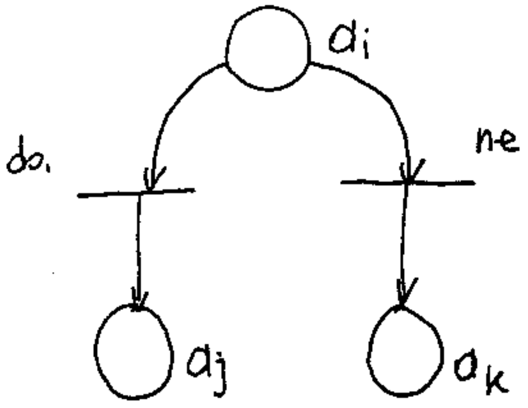
<u>Diagram lahko napišemo v obliki prehajanja stanj:</u>





<u>Vejitev podobno:</u>





Pogledamo bloke, ki so za nas zanimive

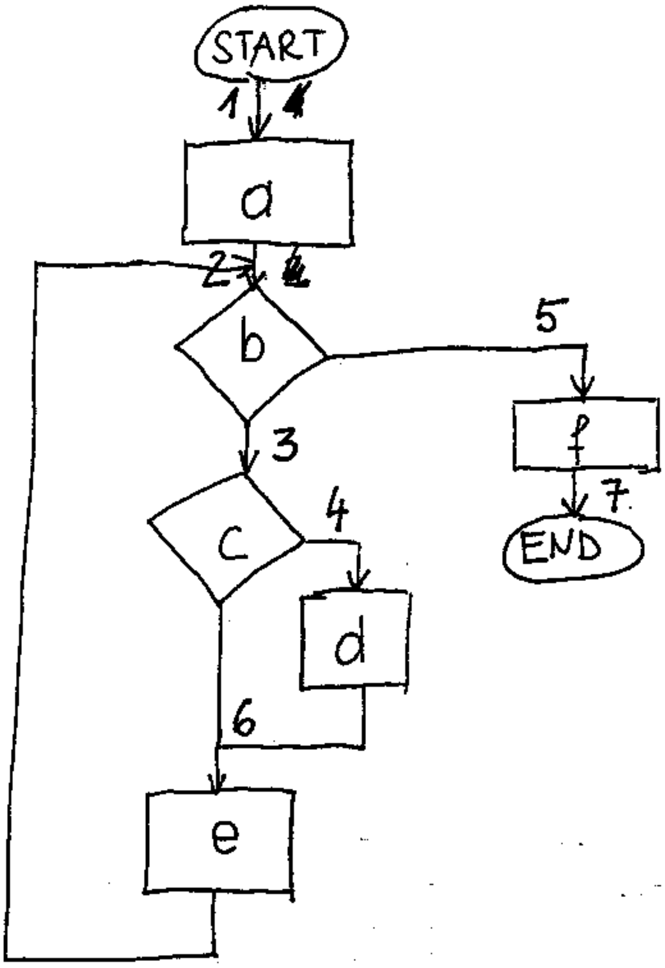
```
Program PN;
var
    y1 : integer;
    y2 : integer;
    y3 : integer;
begin
1⇒  read (y1);          } blok inštrukcij ki se izvede   a
    read (y2);
    y3 := y1;
2⇒ while y1=0 do
     begin
3⇒     if odd (y1) then
       begin
4⇒       y3 := y3 * y2;      } d
         y1 := y1 - 1;
       end;
6⇒     y2 := y2 * y2;        } e
       y1 := y1 - 2;
     end
5⇒  write (y3);             } f
7⇒ end.
```

Napišemo še enkrat

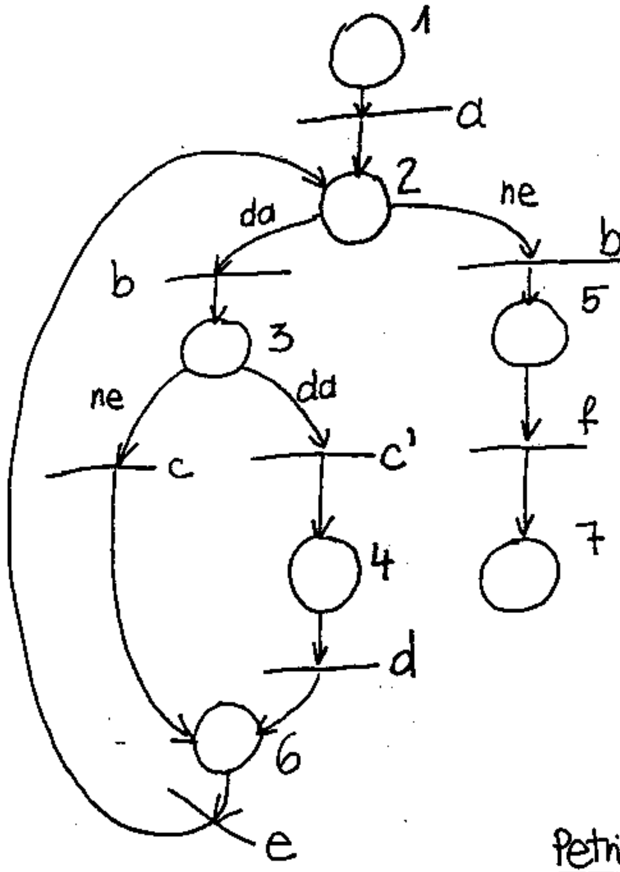
```
begin
   a;
   while b do
     begin
       if c then begin d; end;
     end e
   f
end.
```

V detajle se tu ne bomo spuščali

Narišemo digram prehajanja stanj za poenostavljen program



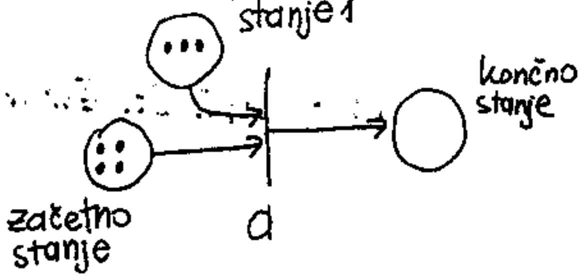
Akcije, vsi ti dogodki so prehajanja! Mesta so prehodi. 1 bo označevala začetek izvajanja programa, ipd.



Petrijev graf

Jezik petrijevih mrež: je jezik, ki ga graf razpozna.

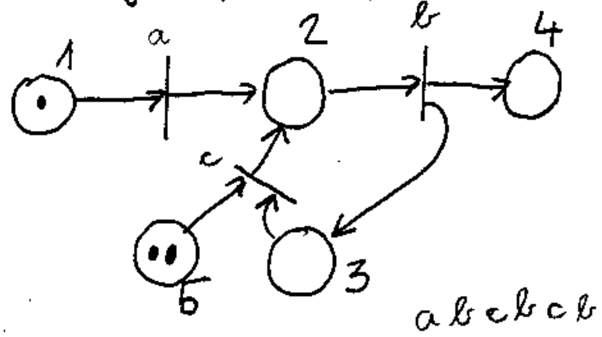
Enostaven primer:



Črka je prehod skozi prehajanje.

→ beseda petrijeve mreže v tem našem primeru je aaa.

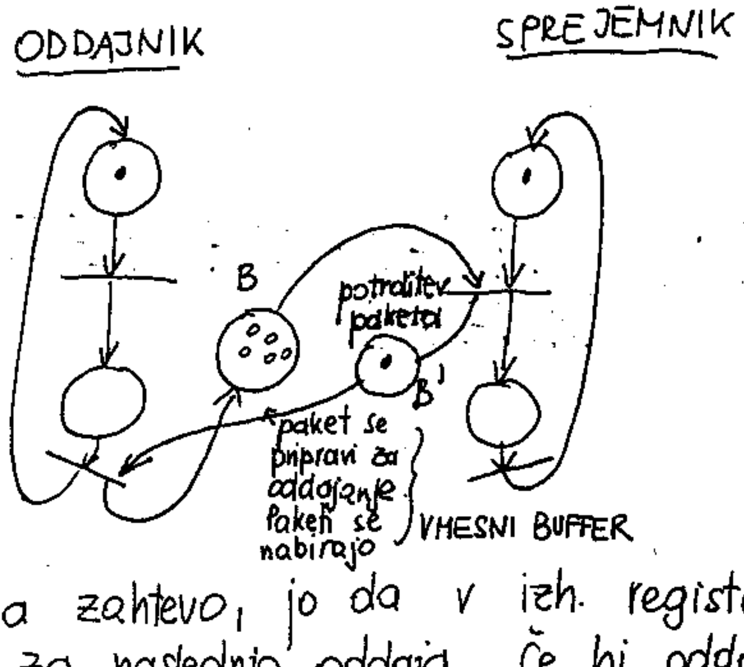
Malce bolj kompleksen primer:



abcbcb
je beseda, ki jo generira ta petrijev graf.

Lahko naredimo petrijev graf, ki generira določeno besedo.
Jezik p.m. je pomemben zato, da vidimo kakšna je moč petrijeve mreže.

(*) Primer za enostavno komunikacijo (enosmerno) med dvema računalnikoma!



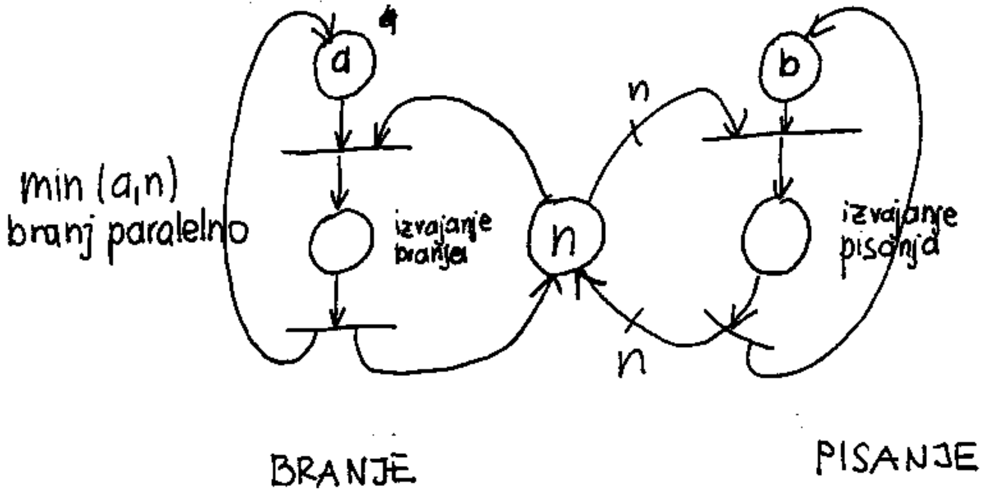
HANDSHAKING
potrjevanje prejetja paketov
zelo počasno. Običajno se pošlje po več paketov skupaj

Oddajnik ima zahtevo, jo da v izh. register in jo odda, pripravi se za naslednjo oddajo. Če bi oddajnik prehitro oddajal, bi se začeli paketi nabirati. Če jih je preveč, se začnejo izgubljati.

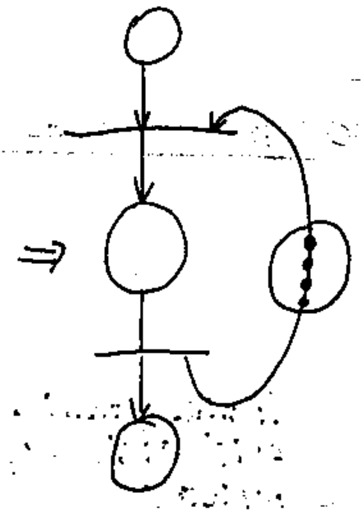
(*) Primer dostopanja do baze podatkov

Če naredimo bazo podatkov, ga ne smejo pisati npr. trije naenkrat. Ko upravljamo z bazo podatkov zaklenemo dostop drugim uporabnikom.

a,b - zač. št. žetonov
n - paralelnih puščic



Ko ni v teku nobenega branja, je pisanje možno, sicer ne. Pišemo lahko samo enkrat.



omejitev na število procesov

↖ ne moremo izvajati več kot 4 paralelne procese hkrati

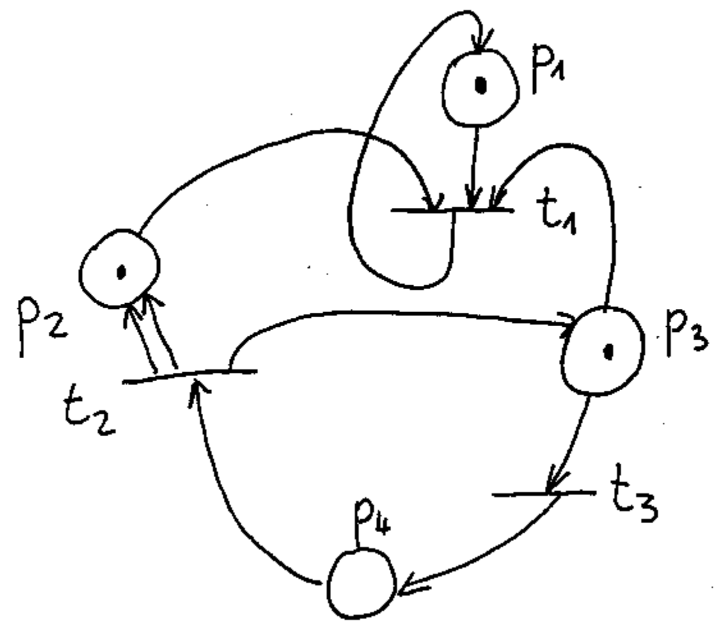
ANALITIČNO REŠEVANJE PETRIJEVE MREŽE

$C = (P, T, I, O)$ $\#p_j$ - število prehodnih mest

$$I = [\underbrace{\#(p_j, I(t_i))}_{\text{število prehodnih mest za prehod } t_i}] = [d_{ij}^-] = D^-$$ vhodna matrika

$$O = [\#p_j, O(t_i))] = [d_{ij}^+] = D^+$$ izhodna matrika

Primer petrijevega grafa



IZPIT!

Določili bomo matriki D^+ in D^-

kolikokrat je p_1 vhoden za t_1

-||- izhoden -||-

$$D^- = \begin{array}{c} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array} \qquad D^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

tukaj štejemo vhodna mesta p_1 p_2 p_3 p_4

tukaj štejemo izhodna mesta

$e_j = [0, 0, \dots, 1, \dots, 0]$ enotni vektor j
1 je na j-tem mestu

$$\boxed{\sigma \geq e_j D^{-1}}$$

$e_j D^{-1}$ - vrstica za t

Primer za t_1

$$\underbrace{[1,0,0]}_{1 \times 3} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{4 \times 3} = [1\,1\,1\,0] \qquad \sigma \geq [1\,1\,1\,0]$$

$\rightarrow 1 \times 4$ pogoj za vžig

$$\boxed{\sigma' = F(\sigma, t_j) = \sigma - e_j D^{-1} + e_j D^+}$$

$D = D^+ - D^-$ Naslednje stanje lahko sedaj napišemo kot:

$$\boxed{\sigma' = \sigma + e_j (D^+ - D^-) = \sigma + e_j D}$$

D pove koliko žetonov v nekem mestu dodajam in odvzemamo ko se izvaja petrijeva mreža

$$\left.\begin{array}{l} D^- = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\[2ex] D^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}\right\} D = D^+ - D^- = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

seštevamo in odštevamo
istoležne elemente

⊛ Primer izvajanja

$\sigma = [1, 1, 1, 0]$

$\sigma \geqslant e_1 D^-$ potem to vžge

$\sigma \geqslant e_1 D^+ = [1\ 0\ 0] \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \underbrace{[1\ 1\ 1\ 0]}_{\text{vektorja sta enaka}}$ ✓ pogoj je izpolnjen

$\sigma \not\geqslant e_2 D^- = \underline{[0\ 0\ 0\ 1]}$ X

↑ tu je manjši, zato pogoj ni izpolnjen, t_2 ne vžge

$\sigma \geqslant e_3 D^- = \underline{[0\ 0\ 1\ 0]}$ ✓ pogoj je izpolnjen

Lahko si izberemo t_1 kot vžig, nov ozn. vektor:

$$\sigma' = \sigma + e_1 D = [1\ 1\ 1\ 0] + [1\ 0\ 0] \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$= [1\ 1\ 1\ 0] + [0, -1, -1, 0] = \underline{\underline{[1\ 0\ 0\ 0]}}$$

⊗ Primer

5 mest

mesto 2 2x izhodno za prehod t_1

$$D^- = \begin{array}{c|c|c} & \begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array} & \\ & \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2 & 0 \end{array}\right] & \begin{array}{c} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{array} \end{array} \quad D^+ = \begin{bmatrix} 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \sigma = [1\ 0\ 0\ 0\ 0]$$

5 prehodnj

matrika vhodov

matrika izhodov

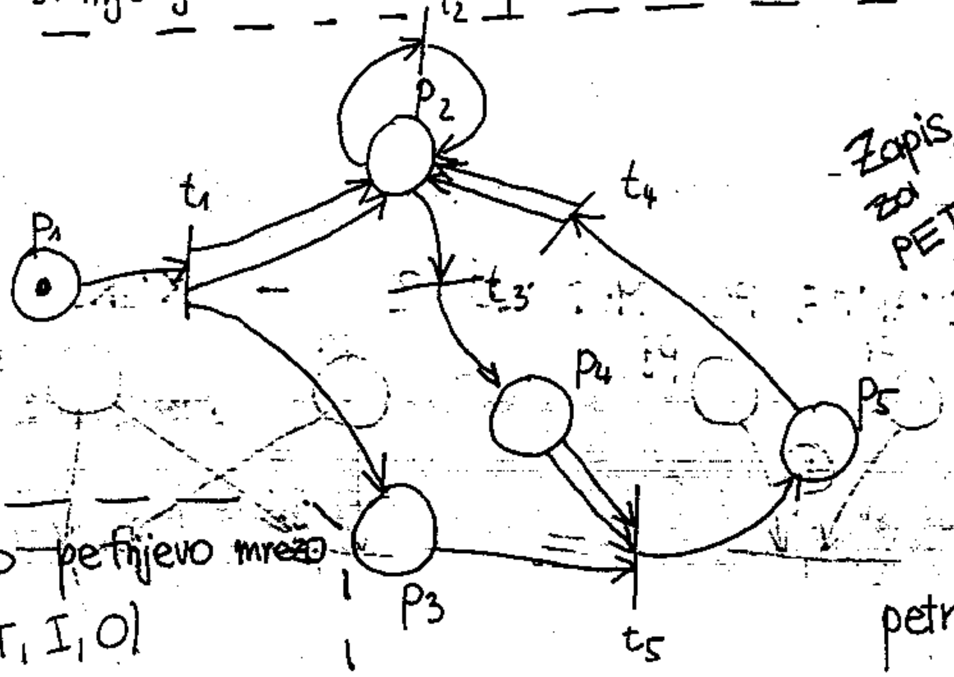
Zapis enostaven za računalnik

ena vrstica opisuje prehod

1 0 0 0 0

↑

prvo stanje je vhodno za 1. prehod



Zapis enostaven za človeka PETRIJEV GRAF

petrijev graf

Napišemo petrijevo mrežo

$C = (P, T, I, O)$

$P = \{p_1, p_2, p_3, p_4, p_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$

$I(t_1) = \{p_1\}$ $\quad O(t_1) = \{p_1, p_2, p_3\}$

$I(t_2) = \{p_2\}$ $\quad O(t_2) = \{p_2\}$

$I(t_3) = \{p_2\}$ $\quad O(t_3) = \{p_4\}$

$I(t_4) = \{p_5\}$ $\quad O(t_4) = \{p_2, p_2\}$

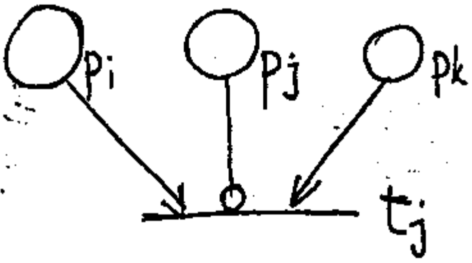
$I(t_5) = \{p_3, p_4, p_4\}$ $\quad O(t_5) = \{p_5\}$

Formalni zapis PETRIJEVA MREŽA

$$D = D^+ - D^- = \begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & -2 & 1 \end{bmatrix}$$

D - pove kakšno bo stanje po užigu

INHIBICIJSKI VHOD



Inhibicijski vhod nam onemogoči vžig

Pogoj, da je prehod t_j izbran

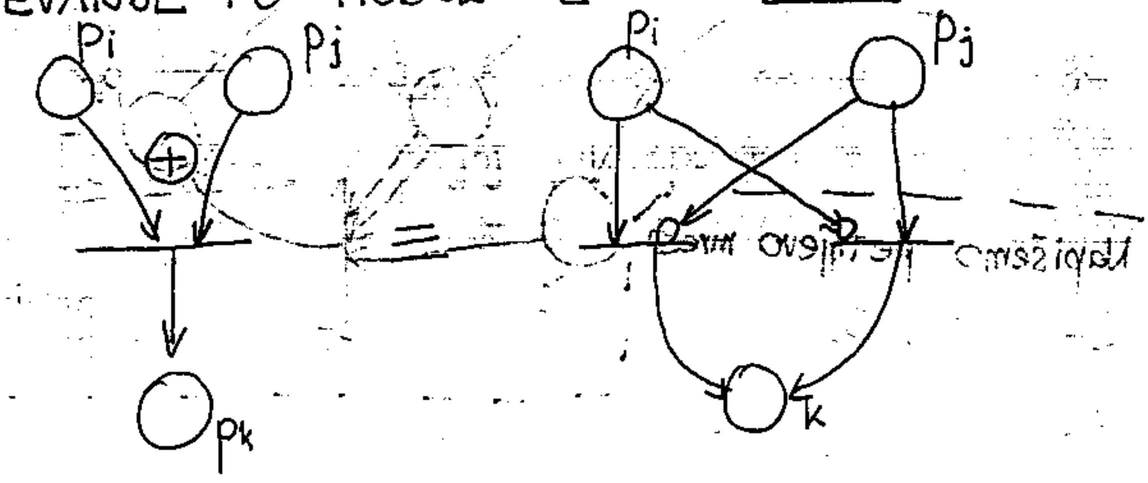
$O(p_i) \geq 1$

$O(p_k) \geq 1$

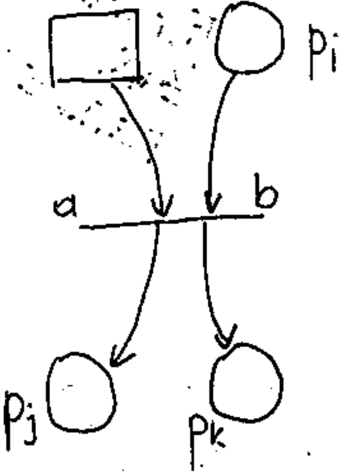
$O(p_j) = 0$

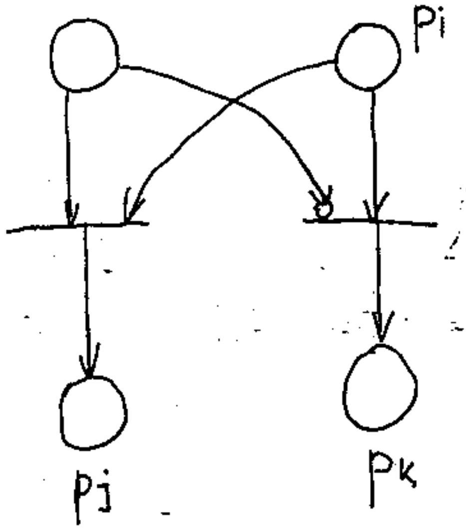
V običajnem modeliranju se temu skušamo izogniti, ker nam ruši matrike in sistem računanja

SEŠTEVANJE PO MODULU 2 - XOR



PREKLOPNIŠKI VHOD - STIKALO



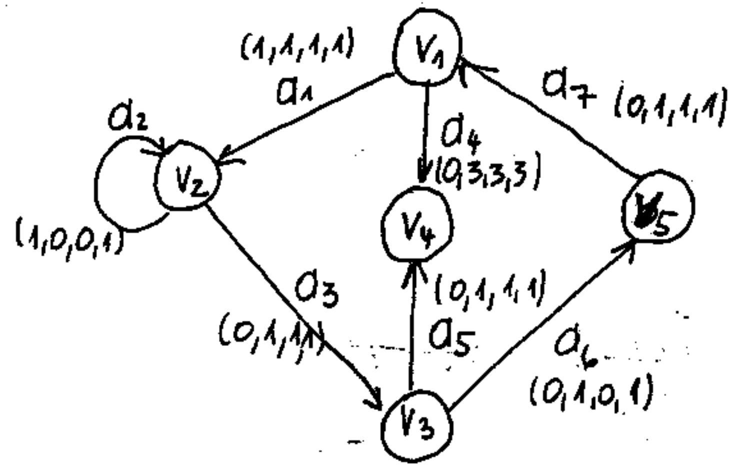


Če je žeton v a vžge leva stran, sicer desna stran

PORAZDELJENO PROCESIRANJE

lahko opišemo z grafom

Primer **Krop-Miler**jevega grafa:



Krop-Milerjev graf

Ponazarja porazdeljeno procesiranje. To bomo kasneje modelirali s petrijevo mrežo

Značilnosti grafa

V – množica procesorjev (spojišč) $V = \{v_1, v_2, v_3, v_4, v_5\}$

usmerjene povezave puščic: pošiljanje procesiranih podatkov iz ene enote v drugo $a = \{v_i ; v_j\}$

Vsaki povezavi ustreza določen četvorček $V_{ij} = (I_{ij}, V_{ij}, W_{ij}, T_{ij})$

I_{ij} – število zahtev v vrsti

V_{ij} – število podatkovnih enot (zahtev) ki se postavijo na izhod (po izvrstitvi)

W_{ij} – število zahtev, ki se odstranijo iz č. vrste, ko se sprocesirajo

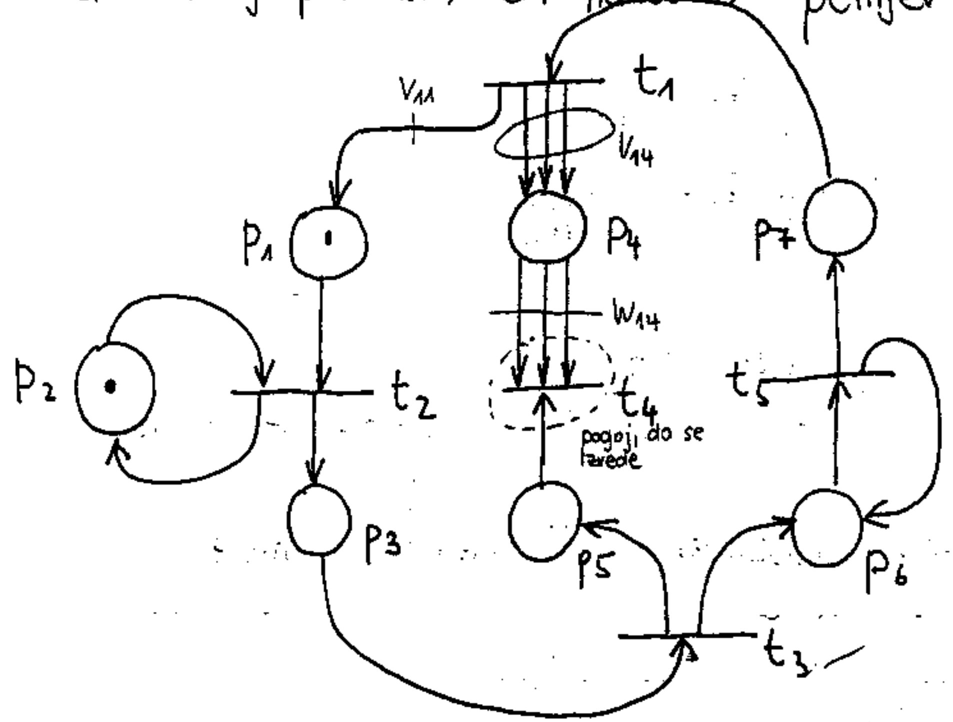
T_{ij} – minimalno število zahtev, ki morajo biti v čakalni vrsti, da se začne procesiranje (koliko zahtev mora biti v vrsti)
- smiselni pogoj je tisti, da je $T_{ij} \geq W_{ij}$

$W_{ij} \leq T_{ij}$

Po procesiranju: $I_{ij}' = I_{ij} - W_{ij}$

po drugi strani: $I_{ij} + V_{ij}$

Imamo dovolj podatkov, da narišemo petrijev graf

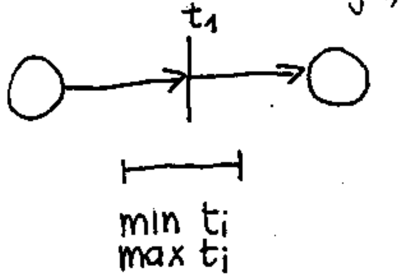


V14 - št. zahtev, ki se izvrš. prog. postavijo v č. vrsto

S simulacijo lahko vidimo kaj se dogaja.

ČASOVNE PETRIJEVE MREŽE

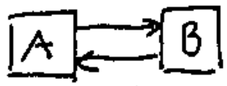
Za vsak prehod t_i bomo določili minimalni čas t_i - kdaj najprej se izvrši določena akcija, in maksimalni čas t_i



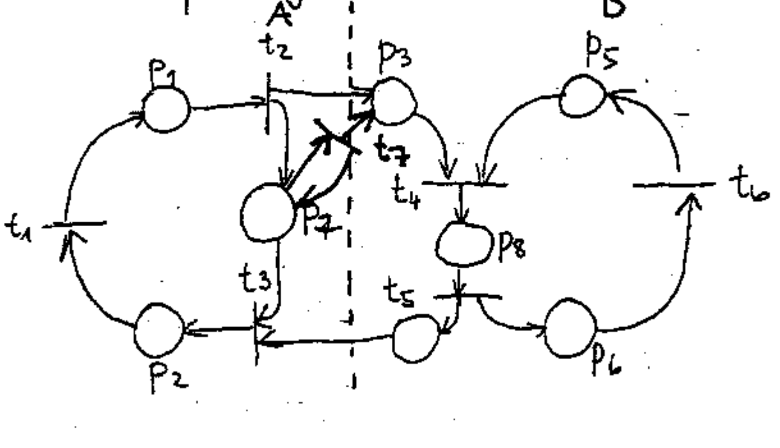
Primer: komunikacijski protokoli. Nekaj časa čakamo na odgovor, ne pa pošljemo zahteve eno za drugo

Odvisno od nastavitve protokola uporabljamo minimalne, oz. maksim. čase.

Primer dveh računalnikov, ki medsebojno komunicirata

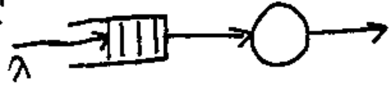


Rač: A pošlje paket B, B vrne odgovor



GPSS General Purpose Simulation System

Primer:



Simulacijski jezik

REALIZIRANJE PRIHAJANJA ZAHTEV

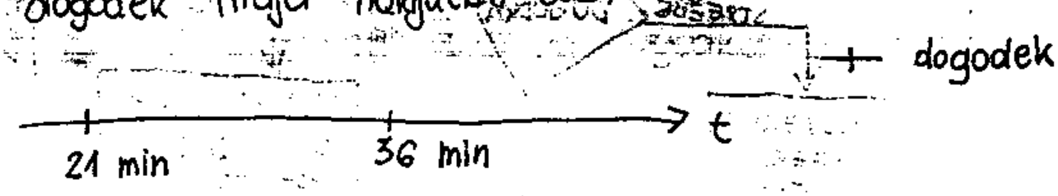
λ 12 .... 24 min Za vse čase bo približno enako število zahtev (pri enakomerni porazdelitvi)

→ 9... 15 min —||— strežnik - verjetnost časa je enaka

→ neskončna vrsta

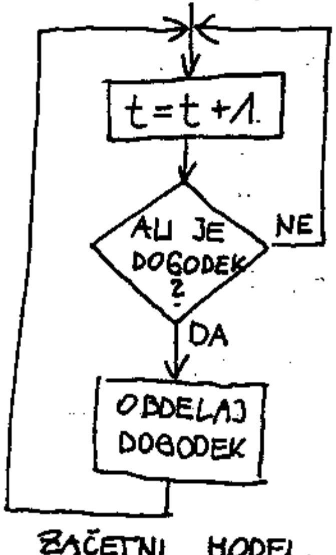
→ nobenih dogodkov (prazna vrsta, strežnik) na začetku simulacije

Definirati moramo dogodke. Ko pride zahteva v vrsto in v strežbo. Primarni - osnovni so vezani na neka naklj. generir. zahtev Sekundarni dogodki so vezani na osnovne, primarne dogode. Čas med dvemi prihodi je ~~naključni~~ primarni in čas strežbe naključni sekundarni dogodek (traja naključen čas)

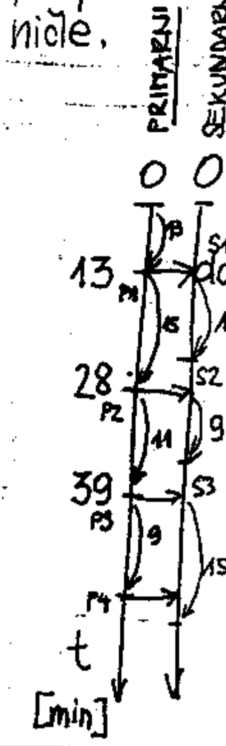


V sistemu, računalniku imamo svoj (ponavadi diskreten) čas, torej je enota sekunda, ura, minuta, ipd. Čas nam torej teče v korakih. V simulacijah vedno teče od ničle.

Simulacija takega programa



ZAČETNI MODEL



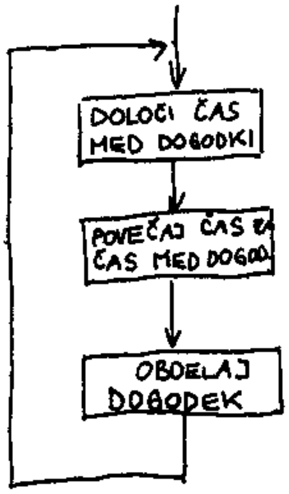
dogodek damo v vrsto in ga obdelam

P_1

$P_1 S_1 P_2$

$P_1 S_1 P_2 S_2 P_3 \ldots$

P_1 proži S_1 in P_2
P_2 proži S_2 in P_3

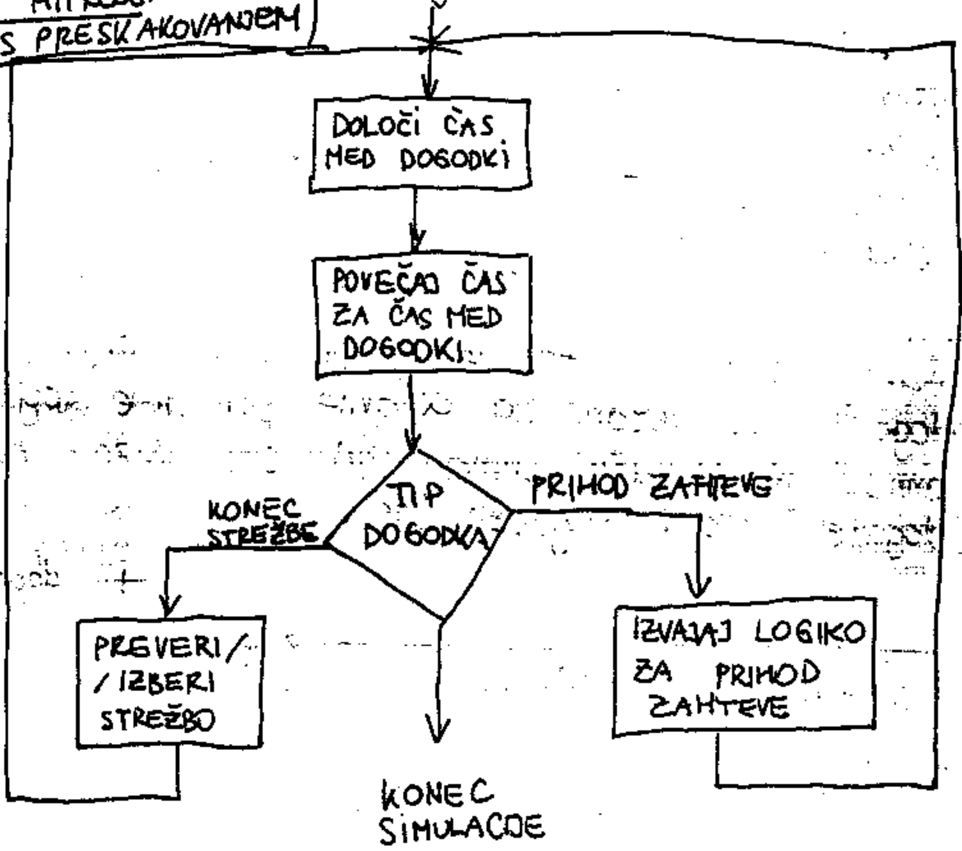


Imamo vrsto dogodkov, ki jih je potrebno obdelati. Čas dejansko lahko postavimo naprej. Ni potrebno, da čakamo takrat, ko se nič ne zgodi.

Rezultati so praktično isti.

Zanko je potrebno končati: Izberemo si dogodek iz liste dogodkov, ki pomeni konec simulacije

BISTVENO HITREJŠI MODEL (S PRESKAKOVANJEM)



MODEL S PRESKAKOVANJEM IN IZHODOM IZ ZANKE

50 min (STOP) primarni dogodek, kjer se simulacija ustavi.

$S = 50$ min
$P_1 = 14$ min

12…24 min – čas med prihodi
9…15 min – čas strežbe

0 → 14 min čas povečamo za 14 min

14 min: → $P2 = 24$ min rand (12,24) (10 + 14)
$S2 = 27$ min (13 + 14) rand (9,15)
$S = 50$ min

} to imamo v vrsti

---

14 min → 24 min

24 min: $P_3 = 44 \sim (20 + 24)$ min
$S_3 = 33 \sim (9 + 24)$ min
→ $S_2 = 27$ min
$S = 50$ min

---

27 min: → $P_3 = 44$ min
$S_3 = 33$ min
$S = 50$ min

---

33 min: → $P_3 = 44$ min
$S = 50$ min

---

44 min: $P_4 = 56$ min $(12 + 44)$
$S_4 = 59$ min $(15 + 44)$
→ $S = 50$ min

---

50 min: STOP

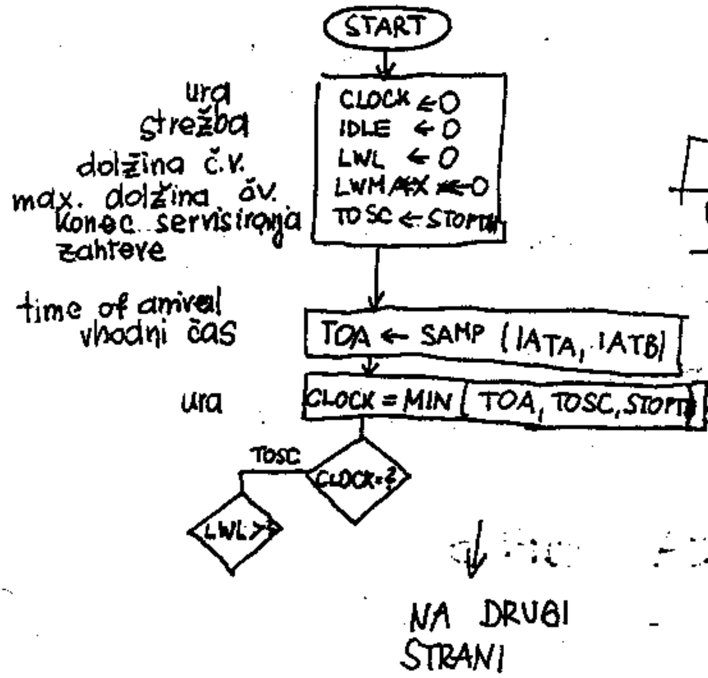
V šestih korakih smo dejansko simulirali 50 minut

Kako pridemo do enakomerne razporeditve.

$\underbrace{12 + RND}_{(0;1)} * 12$ ← verjetnost vsakega dogodka je enaka.

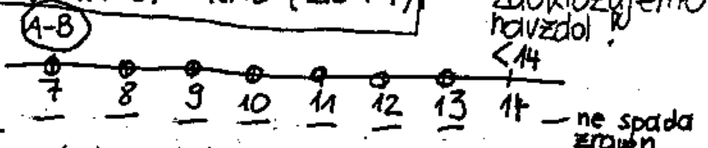
Petek, 18. november 1998

Shema:



$A \pm B$
A … povprečna vrednost
B … odmik primer 10 ± 3 (7, 8, …, 13)
RANDOM 0, … 0,9999… < 1

$$\boxed{T = (A-B) + RND\,(2B+1)}$$ – zaokrožujemo navzdol!

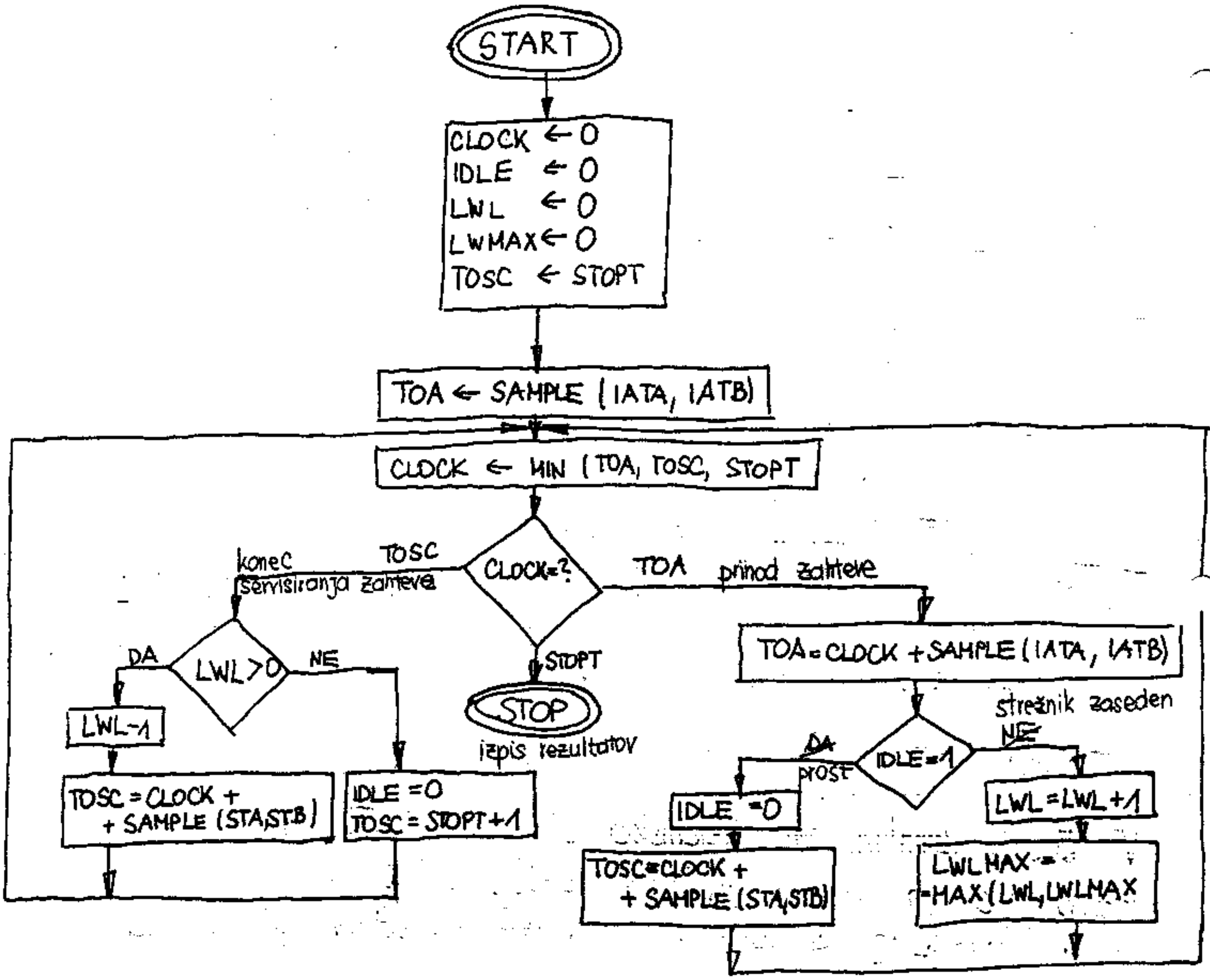


$p(7) = p(8) = \ldots = p(13)$ enakomerna porazdelitev

števila od 7.999…, do 13.9999
⇓ 7 ⇓ 13

STA – povprečni čas strežbe
STB – odstopanje

idle=1 strežna enota je prosta



CLOCK ... ura
IDLE ... status strežnika (0 je prost, 1 je zaseden)
LWL ... dolžina čakalne vrste (length of waiting line)
LWMAX ... največja dolžina čakalne vrste
TOSC ... konec servisiranja zahteve (time of service completed)
STOPT ... konec izvajanja simulacije
TOA ... vhodni čas (time of arrival).
IATA ... povprečni čas prihajanja zahtev (incoming arrival of transaction)
IATB ... odstopanje časa prihajanja zahtev
STA ... povprečni čas strežbe (service time A)
STB ... odstopanje povprečnega časa strežbe (service time B)

BLOK SHEMA DELOVANJA GPSS

Osnovni konstrukti programa GPSS

GPSS je interpreter, ne pa prevajalnik.

Transakcija se ubije, ko pride do enega izmed pogojev TOSC, STOPT, TOA
Ura je najpomembnejši del, programer je tisti, ki da uri smisel. GPSS je generator dogodkov, model za obdelavo dogodkov.

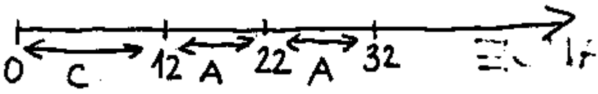
BLOK GENERATE

Privzete vrednosti

A : 0
B : 0
C : 0
D : ∞
E : 0

A ± B
Zahteve so enakomerno razporeditev
Privzeta (DEFAULT) vrednost A in B je nič

C začetni interval - offset odmik

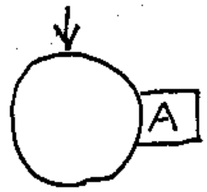


D število zahtev, ki jih zgeneriramo, potem se generator ustavi

E S kakšno prioriteto označimo transakcijo 0...127. Pomembno pri čakalnih vrstah → se postavi na začetek čakalne vrste

BLOK TERMINATE

Transakcija umre v bloku TERMINATE. Parameter A nam pomeni števec zahtev, ki je za vse skupni. Ko pride števec na nič, se simulacija konča. Privzeta vrednost za A je nič, blok ne vpliva na števec!

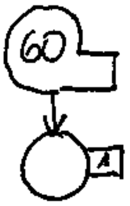


$TS = TS - A$

A: 0

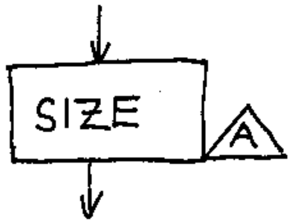
V celi simulaciji ima <u>samo en števec!</u>

Primer :



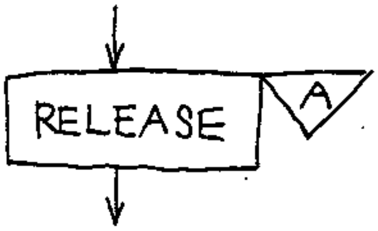
Po 60 urnih enotah pride TS do nič, torej konec simulacije

BLOK SIZE IN RELEASE

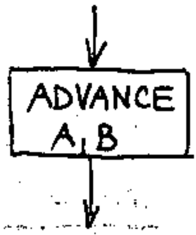


A ime strežnika
Ko pride zahteva, nam zasede (SEIZE) strežnik, dokler ne pridemo do bloka (RELEASE).

Če je strežnik zaseden, zahteva ne more iti do njega.



BLOK ADVANCE

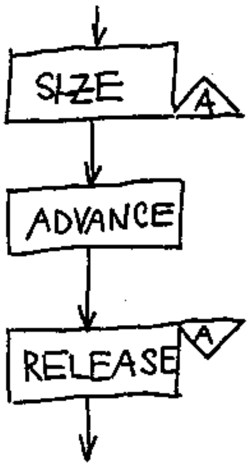


Blok advance je dejansko zakasnitev, ki jo ponazarjamo s časom strežbe

$T = A \pm B$

Privzete vrednosti za A in B je nič.

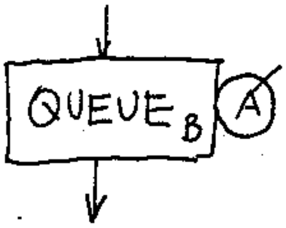
Tipičen primer strežnika je kombinacija zgornjih treh blokov.



Če transakcija pride do zasedenega SIZE bloka, čaka na sprostitev.

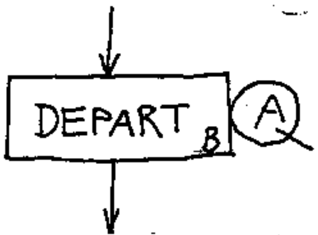
ČAKALNE VRSTE

Nam omogočijo analizo transakcij in vrednotenje simulacije. Dajo informacijo o tem, kaj se v sistemu dogaja.



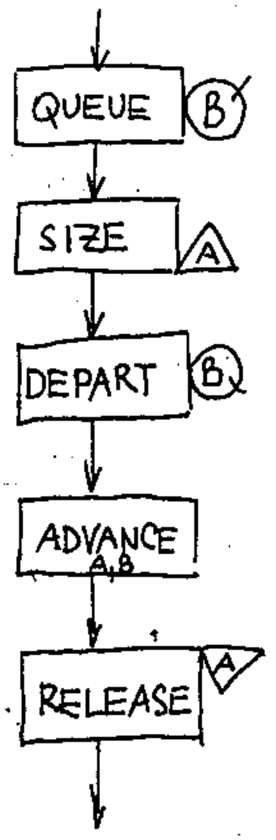
A ime čakalne vrste

vstopanje v čakalno vrsto



Transakcija ima svojo uro

izstopanje iz čakalne vrste



Parameter B : Koliko prostora zavzame zahteva v vrsti.

Primer: Pride v vrste element, ki je večji kot običajno

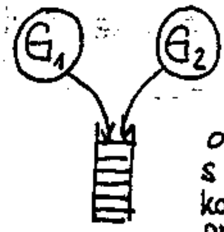
V splošnem je lahko ena transakcija element več čakalnih vrst

Primer BRIVCA

striženje 35 ± 10 minut
striženje in britje 60 ± 20 minut

$X_S = 18 \pm 6$ min
$X_B = 10 \pm 2$ min

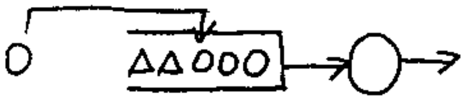
8ur

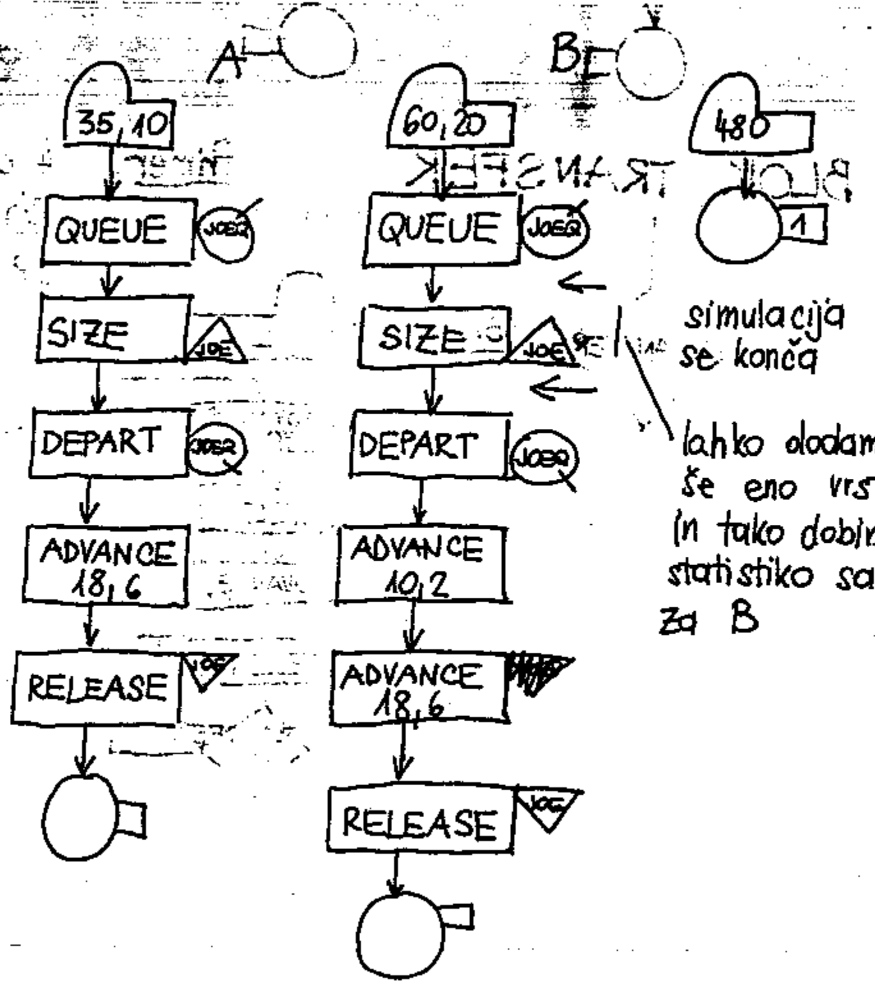


opremi se s časom in kateri vrsti pripada

Prioritete

O višja prioriteta 2
Δ nižja prioriteta 1





simulacija se konča

lahko dodamo še eno vrsto in tako dobimo statistiko samo za B

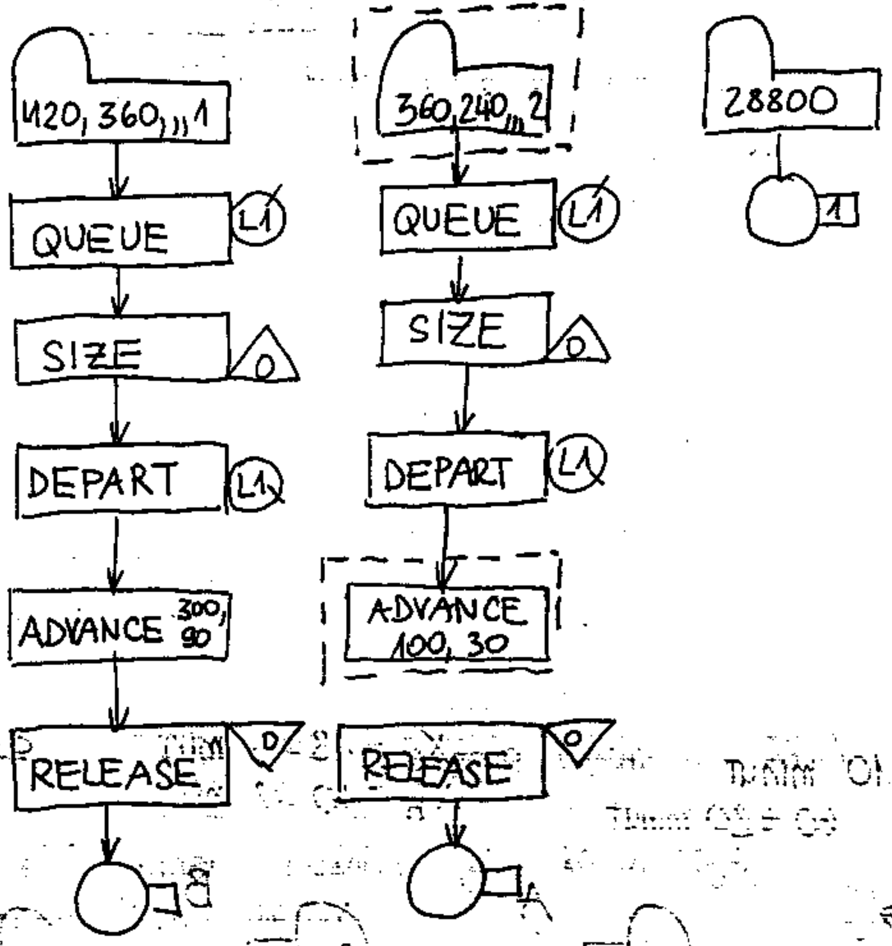
striženje

Striženje in britje

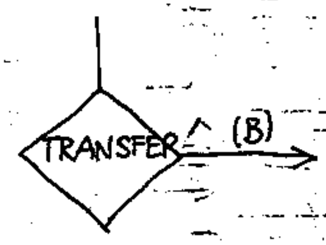
delovni čas

Primer uporabe prioritete v avtomehanični delavnici

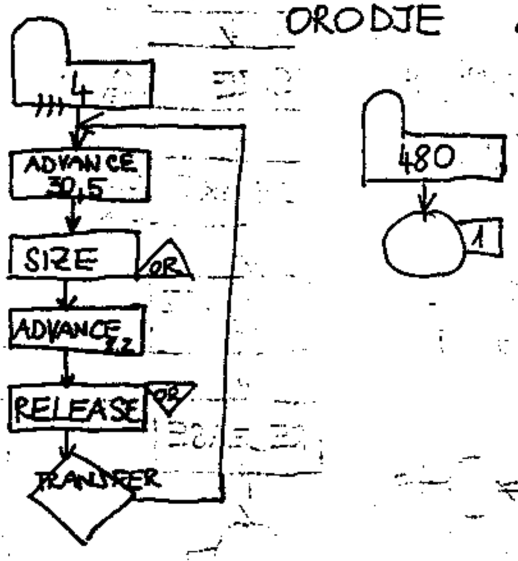
1	420 ± 360	300 ± 90
2	360 ± 240	100 ± 30



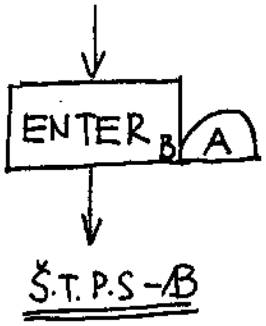
BLOK TRANSFER

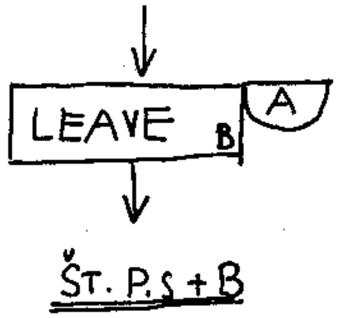


Primer: 4 delavci
30 ± 5 minut za sestavljanje
ORODJE 8 ± 2 minut



PRIMER Z VEČ STREŽNIKI - BLOK ENTER, LEAVE

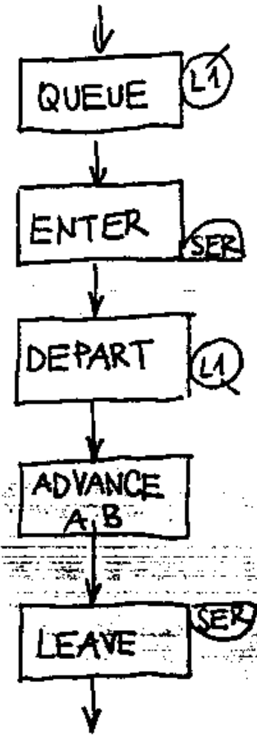


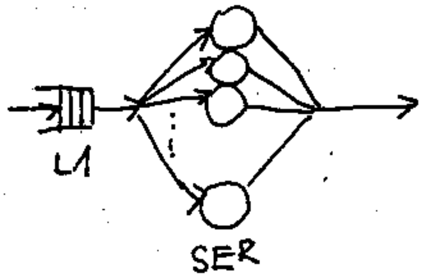


B... koliko strežnikov nam zasede zahteva. Privzeto: 1

A - ime

Primer uporabe





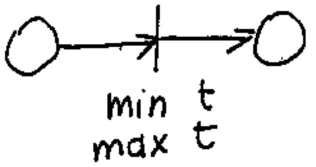
p_1 – blok je pripravljen (paket)
p_2 – paket je bil sprejet
p_3 – informira proces rač A. da je bil blok poslan

A p_4 – paket je bil sprejet
p_5 – računalnik B je pripravljen sprejeti blok
p_6 – rač.B informira svoj sistem, da je bil paket sprejet

p_7 – pravilno poslan
p_8 – pravilno sprejet

HANDSHAKING – rokovanje
Na ta način dela večino komunikacij.

Paketi se lahko zaradi motenj na liniji pokvarijo. Dodati moram dodatno mesto. V p_7 čakamo na potrditev. Mora biti neka časovna zakasnitev. Kdaj lahko vžge?



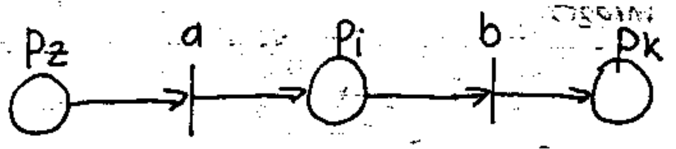
Kakšen bo min t_7 ?

$\underline{\min t_7 > \max t_4 + \max t_5 + \max}$

Takrat bomo ponovili (smo prepričani, da se je paket izgubil)

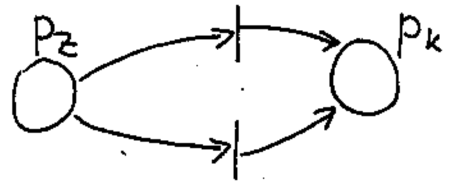
Primer:

zaporedno



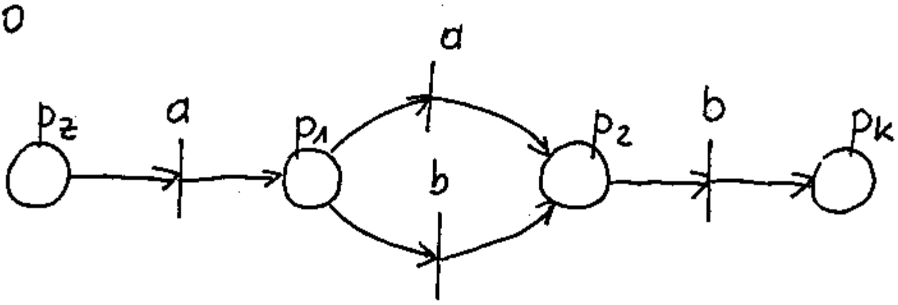
$\underline{L(c_1) = ab}$ .... besede $L(c)$ - jezik

vzporedno

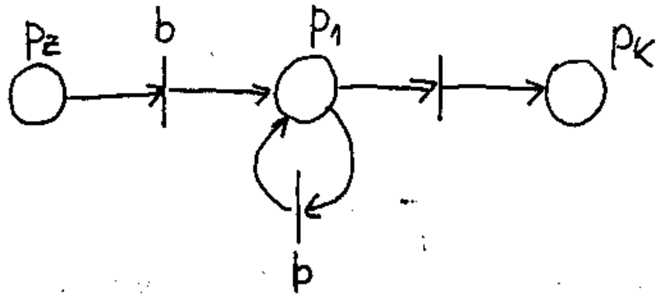


$\underline{L(c_2) = a + b}$

kombinirano

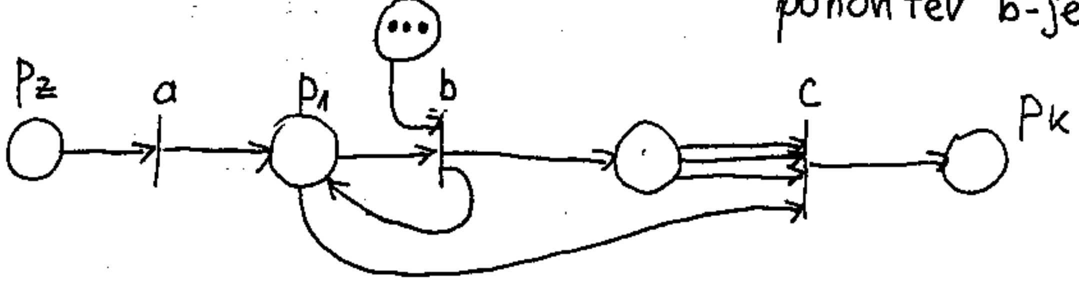


$\underline{L(c_3) = a(a+b)b}$ jezik
$\{aab\}$
$\{abb\}$ besede (vse poti, ki so možne)



$L(c_4) = b^+ a$

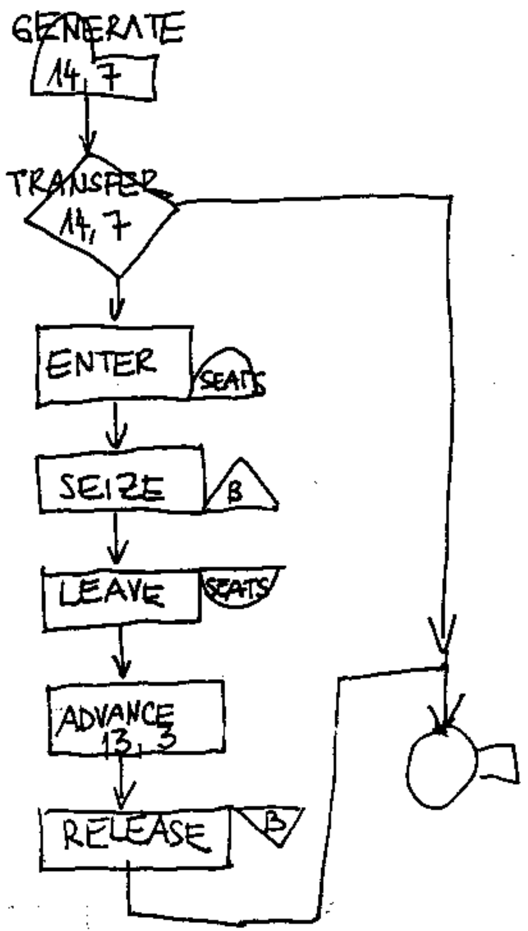
b^+ -ena ali reč ponovitev b-jev



$L(c_5) = abbbc = ab^3c$

- def. pet. mrežo
- def. ozn. petrijevo mrežo
- petrijev graf
- ozn. petrijev graf
- Izvajanje petrijevega grafa, mreže
- primeri uporabe pet. mrež
- matrična oblika zapisa pet. mreže

Primer: Frizerski salon, par stolov

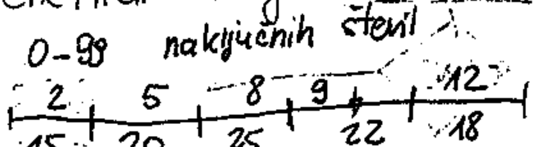


Naključna števila

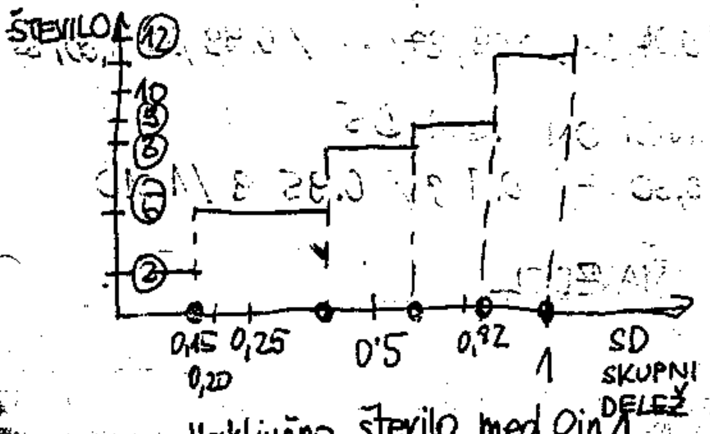
N. Š	R. F	S. D		INT
2	0,15	0.15	0~0.15	1
5	0.20	0.35	0,15+ ~0,35	2
8	0, 25	0.60	0,35+ ~0,60	3
9	0.22	0.82	0,60+ ~0,82	4
12	0.18	1.00	0,82+ ~1,00	5

→ širina intervala

Kako narediti program, ki bo generiral naključna števila?



Slika



Naključno število med 0 in 1

TEST FUNCTION RN1, D5 D5... diskreten interval, opisan s petimi intervali

0.15, 2 / 0.35, 5/ 0.60, 8 / 0.82, 9/1, 12

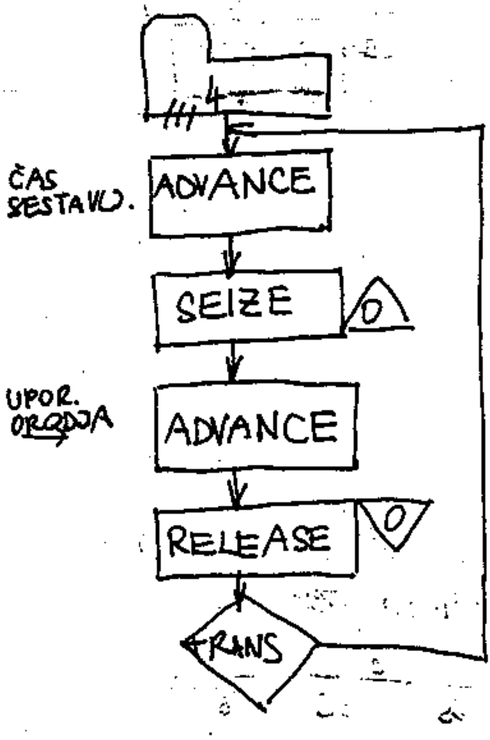
ČAS	DELEŽ	SKUPNI DELEŽ	TBO
2	0,10	0,10	
3	0,30	0,40	
4	0,40	0,80	
5	0,20	1,00	

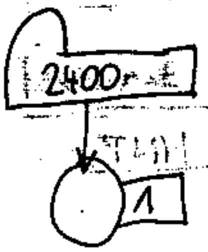
→ uporabili bomo 7. naključni generator

TBO FUNCTION RN7, D4

0.1, 2/ 0.4, 3 / 0.8, 4 / 1, 5

Primer z delavci z orodjem





SEST.			UPOR. ORODJA		
25	0.01	0.01	6	0.05	0.05
26	0.03	0.04	7	0.25	0.30
27	0.05	0.09	8	0.40	0.70
28	0.10	0.19	9	0.25	0.95
29	0.18	0.37	10	0.05	1.00
30	0.26	0.63			
31	0.18	0.81			
32	0.10	0.91			
33	0.05	0.96			
34	0.03	0.99			
35	0.01	1.00			

SES FUNCTION RN1, D11

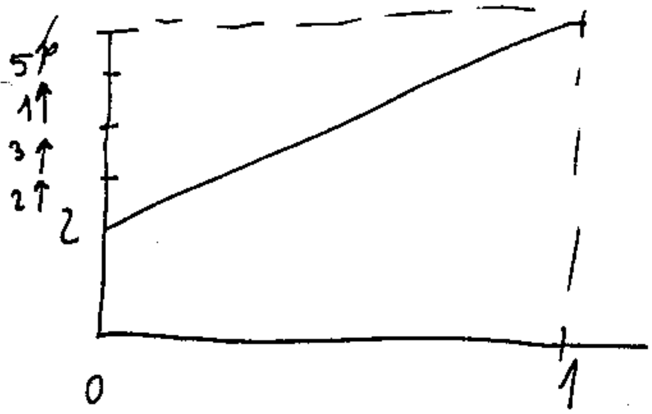
0.01, 25/ 0.04, 26 / 0.09, 27/ . . . / 0.99 34/1,00/35

ORO FUNCTION RN1, D5

0.05,6/ 0,30,7/ 0.7,8/ 0.95, 9 /1,10

VEDNO SE ZAOKROŽUJE NAVZDOL
(CELI DEL ŠTEVILA)

ZVEZNE FUNKCIJE

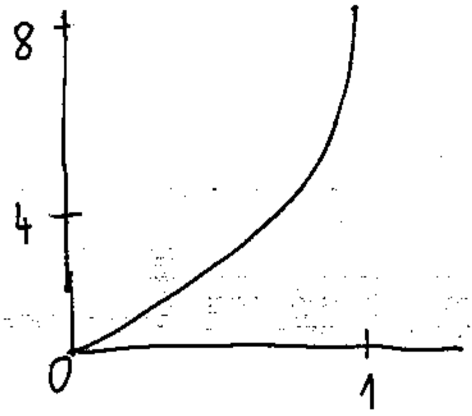


Napišemo tako funkcijo dva param

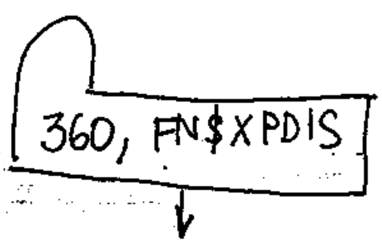
LF FUNCTION RN2, C2
0,2 / 1,6 –vmes so linearne vrednosti

Hočemo enakomerno porazdeljeno števila 200 - 375

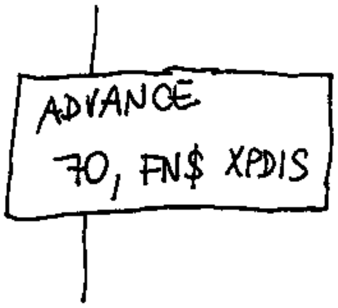
FAST FUNCTION RN6, C2
0,200 / 1,376 ← zaradi zaokroževanja navzdol



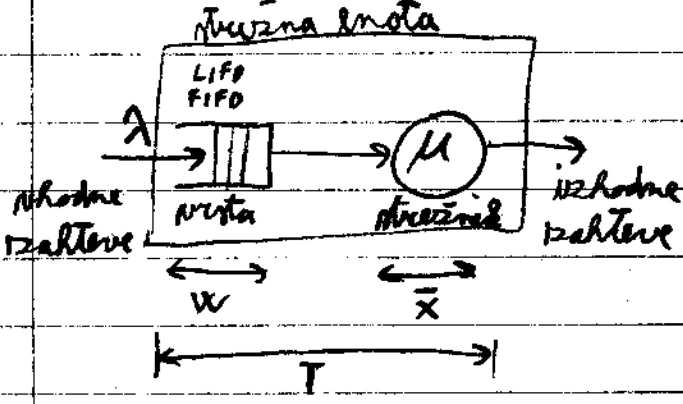
XPDIS FUNCTION RN1, C24
0,0 / 0,1, 0.104 / 0.2, 0.22 / . . .



Imamo 360 · vrednost funkcije
Lahko dobimo od 0 pa do skoraj neskončno, vendar pa je povprečje pri 360



LITTLOV TEOREM



λ - intenzitivnost prehajanja sistema

μ - intenzitivnost procesiranja zahtev

ρ - uporabnostni faktor strežne enote (SE)

N - povprečno št. zahtev v sistemu

T - povprečni čas bivanja zadeve v sistemu

x̄ - povprečni čas procesiranja ene zahteve

W - čas čakanja v vrsti

p - verjetnost

$\boxed{N = \lambda \cdot T}$ Littlovo pravilo

$\boxed{\mu = \frac{1}{\bar{x}}}$ $\boxed{\rho = \frac{\lambda}{\mu} = \lambda \cdot \bar{x}}$

$\boxed{T = W \cdot \bar{x}}$

$M/M/m/n/k \rightarrow$ velikost populacije

eksponentno porajanje zahtev na vhodu

eksponentno procesiranje

št. strežnikov

dolžina vrste

$N_S = \frac{\lambda}{\mu} = \rho = 1 - p_0 \rightarrow$ št. zahtev v strežniku

$\boxed{N = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}}$ $\boxed{p_0 = 1 - \rho}$ verjetnost praznega sistema

$$T = \frac{N}{\lambda} = \frac{\rho}{\lambda(1-\rho)} = \frac{\lambda}{\lambda(\mu \cdot \lambda)} = \frac{1}{\mu - \lambda}$$

14. naloga: V center mesta vodijo 4 vpadnice. Intenziteta vhodov po posameznih vpadnicah je:

1.) 1500+/-100 vozil na uro
2.) 1200+/-200 vozil na uro
3.) 2300+/-400 vozil na uro
4.) 5000+/-1000 vozil na uro

0,25 0.04

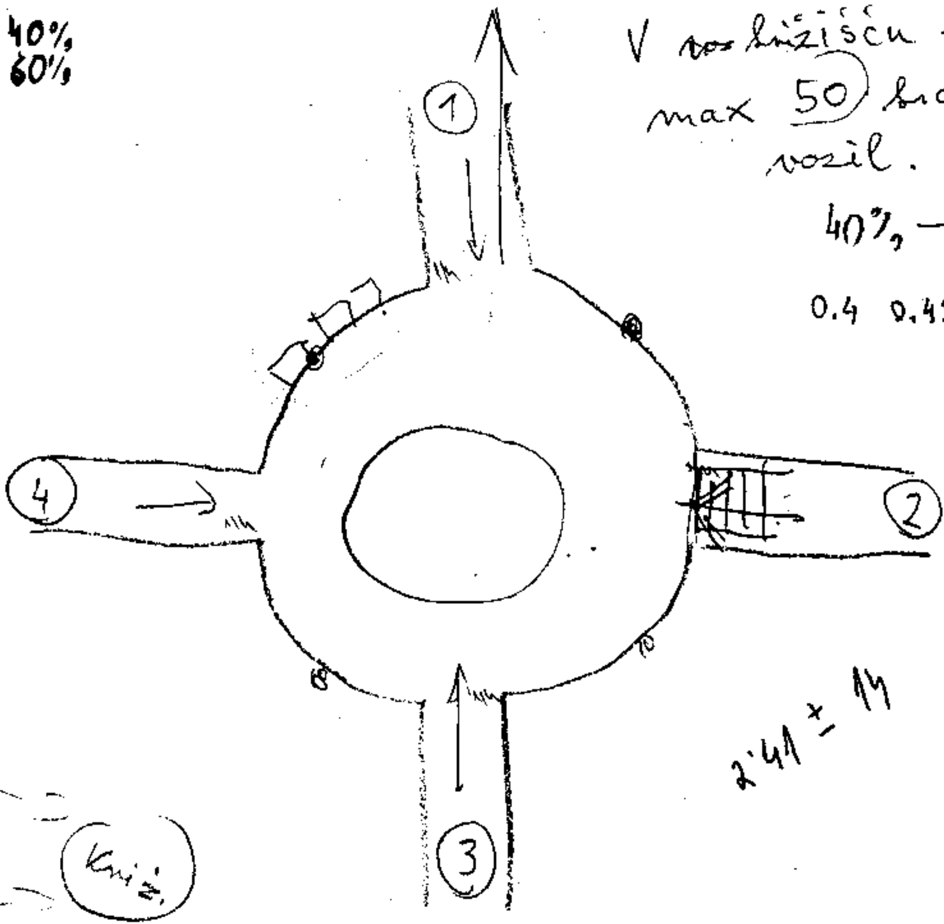
V centru mesta je krožno križišče, v katerega vstopi vsako prišlo vozilo. V bližnji okolici križišča ostane 50+/-10% vseh vozil, ki prevozijo krožno križišče, ostala vozila pa izberejo eno od vpadnic in se po njej odpeljejo iz mesta (možnost izbire vpadnice je enakomerno porazdeljena). Od vstopa do izstopa iz krožnega križišča mine 10+/-2 sekundi. Postavi model navedenega križišča in na koncu komentiraj, kaj storiti s podanimi parametri, da bi se v vrsti pred vstopom v križišče čakalo manj kot 10 sec.

40%
60%

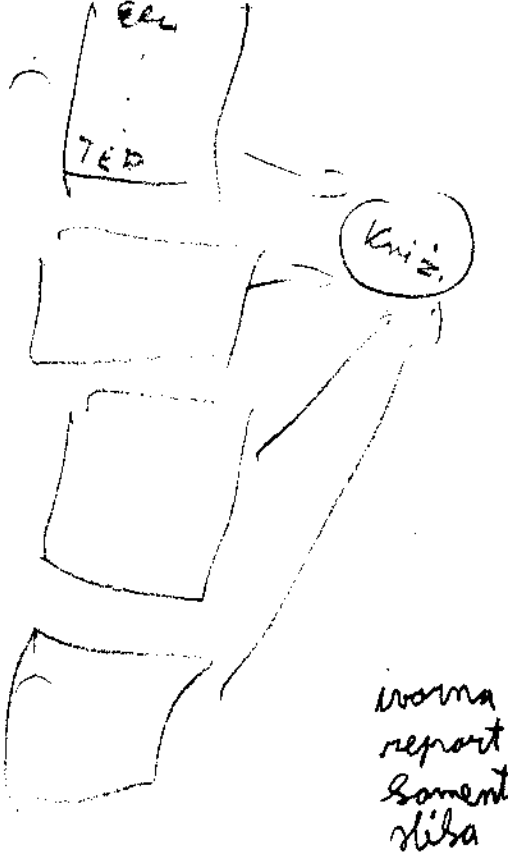
V križišču je lahko max 50 krožečih vozil.

40% — 60%

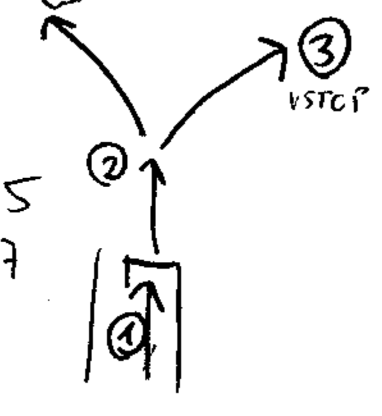
0.4 0.45 0.5 0.55 0.6



2'41 ± 14



3600 : 1500 = 2'4
3600 : 1600 = 2'25
3600 : 1400 = 2'57



ivorna
report
komentar
slika

```
 210          12              QUEUE              53  87         14 27         0
 220          13              ENTER              39  60               0       0
 230          14              DEPART             39  60               0       0
  40          15              TRANSFER           39  60               0       0
 300          16              GENERATE          105 299               0       0
 310          17              QUEUE             105 299         27 88         0
 320          18              ENTER              79 211               0       0
 330          19              DEPART             78 211               0       0
 340          20              TRANSFER           78 211               0       0
 400        AVTO1             LEAVE              73 175               0       0
 410          22              TERMINATE          73 175               0       0
 450        AVTO2             ADVANCE            96 169         12 18         0
 460          24              LEAVE              84 151               0       0
 470          25              TERMINATE          84 151               0       0
 600          26              GENERATE            1   1               0       0

     [Esc] to abort     [SPACE] for next page    [PgDn] to skip
```

```
GPSS/PC Report file REPORT.GPS.  (V 2, # 40390.21875)  12-09-1998 16:  page 3

LINE        LOC            BLOCK_TYPE        ENTRY_COUNT   CURRENT_COUNT   RETRY
 610         27             TERMINATE                  1               0       0


QUEUE          MAX   CONT.  ENTRIES ENTRIES(0)  AVE.CONT.  AVE.TIME     AVE.(-0)  RETRY
 VPADNICA1    9 19   9 19    35 61       2   5 5.65  8.29 891.45 1222.57  1331.73    0
 VPADNICA2   10 14  10 13    38 44       3   3 4.22  6.00 888.87 1227.50  1317.32    0
 VPADNICA3   14 27  14 27    53 87       4   6 5.81 11.70 876.81 1210.75  1300.43    0
 VPADNICA4   27 90  27 88   105 299      8  26 11.50 40.10 875.33 1207.01  1321.96   0
                                                                    948.97   948.39
                                                                    965.06   948.18

STORAGE          CAP. REMAIN.  MIN.   MAX.    ENTRIES AVL.   AVE.C.  UTIL. RETRY DELAY
 KRIZISCE         18       0      0     18       344    1    17.30   0.961    0   147
                  12       0      0     12       169    1    11.42   0.952    0    60


     [SPACE] for another report              Any other key to end
```

```
GPSS/PC Report file REPORT.GPS.  (V 2, # 40390.21875)  12-09-1998 16:  page 2

LINE        LOC            BLOCK_TYPE        ENTRY_COUNT   CURRENT_COUNT   RETRY
 210         12             QUEUE                     87              27       0
 220         13             ENTER                     60               0       0
 230         14             DEPART                    60               0       0
 240         15             TRANSFER                  60               0       0
 300         16             GENERATE                 299               0       0
 310         17             QUEUE                    299              88       0
 320         18             ENTER                    211               0       0
 330         19             DEPART                   211               0       0
```

POROČILA (REPORT.GPS)

1. Osnovne karakteristike:

Start_time: zacetek simulacije
End_time: konec simulacije
Blocks:
Facilities:
Storage: koliko je resoursov

2. Facilities:

Entry_count: št. transakcij, ki so vstopile v vrsto
Courent_count: št. transakcij, ki so obtičale v tem stavku po koncu simulacije

3. Queue:

Max: maksimalno število čakajočih
Count: število čakajočih pa koncu simulacije
Entries: število vhodov v vrsto skozi simulacijo
Entries(0): število transakcij, ki vstopijo v prazno vrsto
Ave. Count: povprečni čas čakanja v vrsti glede na čas transakcije v sistemu
Ave. Time: povprečni čas čakanja v vrsti
Ave(-0): povprečni čas čakanja v vrsti brez transakcije, ki so vstopile v prazno vrsto

4. Storage

Cap.: kapaciteta resoursa
Remain: koliko transakcij je ostalo po koncu simulacije
Min: minimalna zasedenost resoursa
Max: maksimalna zasedenost resoursa
Entries: kolikokrat je bil resours zaseden
Ave. C.: povprečni čas zasedenosti resoursa s strani ene transakcije
Utilization: procent zasedenosti resoursa s strani ene transakcije

@ VAJA-GPS EDIT 100
DISPLAY DELETE 100
END SAVE VAJA&GPS
START 1

UKAZI GPSS

resours STORAGE, N

GENERATE N	Vsakih N časovnih enot skreiraj novo transakcijo.
GENERATE A,B	Če B ni funkcija, se vsakih A±B časovnih enot kreira nova transakcija.
GENERATE ,,,N	Istočasno bo skreiranih natanko N transakcij, potem pa nobena več.
SEIZE A	Transakcija zasede resurs A, če je ta prost; če ni prost, se transakcija postavi v FIFO vrsto in čaka na prostost resursa.
RELEASE A	Transakcija, ki poseduje resurs A, resurs vrne sistemu.
ADVANCE N	Trajanje servisiranja transakcije za N časovnih enot.
ADVANCE A,B	Če B ni funkcija, servisiranje transakcije poteka A±B časovnih enot.
QUEUE A	Vklop tarifiranja statistik za vrsto z imenom A.
DEPART A	Izklop tarifiranja statistik za vrsto z imenom A.
ENTER storage_name,N	Transakcija skuša zasesti N resursov iz skladišča z imenom storage_name.
LEAVE storage_name,M	Vračanje M resursov v skladišče z imenom storage_name.
TRANSFER	Skok transakcije.
TRANSFER ,labela	Brezpogojni skok na vrstico z imenom labela.
TRANSFER .15, labela1, labela2	15% transakcij skoči na lokacijo labela2, ostalih 85% pa na labela1.
TRANSFER BOTH, labela1, labela2	Transakcija se pošilja na stavek labela1, če je ta zasedena se transakcija pošlje na stavek z labelo2, če pa je tudi ta zaseden, se izmenično preverja, kateri stavek ostane prost in tja usmeri transakcijo.

labela function RN1,D5
0.2,2/0.4,4/0.6,6/0.8,8/1.10
ADVANCE FN$labela

ASSIGN - priredi vrednost

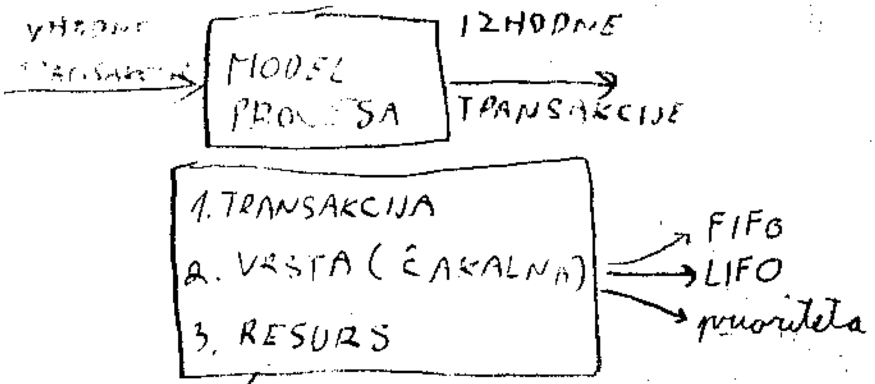
VAJE

OB $7^{4:}$

Virant: Modeli in Simulacije

GPSS

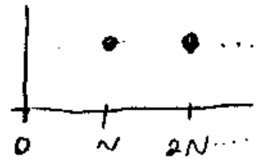
General Purpose Simulation Software



[illegible] lahko
zaseden s strani
največ 1. transakcije

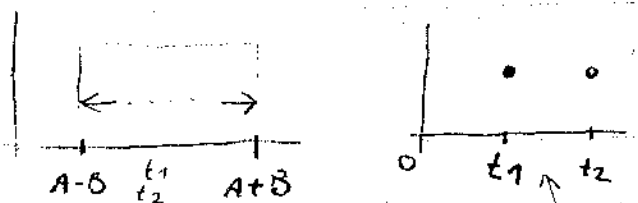
UKAZI GPSS

GENERATE␣N - vsakih N časovnih enot kreiraj novo transakcijo



prva transakcija šele v času N

GENERATE␣A,B – če B ni funkcija, se vsakih A±B časovnih enot kreira nova transakcija



tukaj notri izbere naključni čas, potem pa v tem času uvede transakcijo

GENERATE␣,,,N – istočasno bo kreiranih natanko N transakcij, potem pa nobena več.

max št. transakcij v sistemu

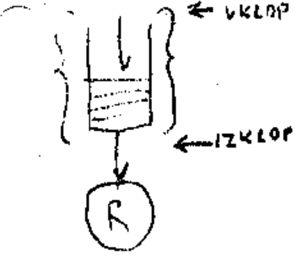
SEIZE␣A – Transakcija zasede resurse A, če je ta prost; če ni prost, se transakcija postavi v FIFO vrsto (za to poskrbi GPSS) in čaka na prostost resursa.

RELEASE␣A – Transakcija, ki poseduje resurs A, resurs vrne sistemu

ADVANCE␣N – trajanje servisiranja transakcije je N časovnih enot.

ADVANCE␣A,B – če B ni funkcija, ~~pro~~ servisiranje transakcije poteka A±B časovnih enot.

QUEUE␣A - vklop tarifiranja statistik za vrsto z imenom A



DEPART␣A — izklop tarifiranja statistik za vrsto z imenom A (sledenje notranjemu predmetu)

Primer: Stranke prihajajo v brivski salon, kjer dela en brivec.

Prihod strank : 16 ± 4 minute

1x britje traja 13 ± 2 minute

Strežba strank : FIFO

Z modelom ponovimo 8 urni delavnik

```
                vsakih
GENERATE␣16,4 //16 č.e. ±4
               nova stranka (paket)
QUEUE␣VRSTA
SEIZE␣BRIVEC  ← DEPART␣VRSTA              B1
ADVANCE␣13,2 //13±2 č.e. je uporaba (μ)
RELEASE␣BRIVEC
TERMINATE
  ↳ transakcija se uniči
```

```
            8x60min
GENERATE␣480        B2
TERMINATE␣1   ← za ∀ transakcijo, ki vstopi: 1. uniči transakcijo
                                              2. TC := TC - TCD
                                              3. if(TC:=0) then stop simulation
  ⇓
TCD:=1 termination counter decrement

START␣1   ← kontrolni stavek
  ⇓
TC:=1
Termination counter
```

ENTER ␣ storage name, N - transakcija skuša zasesti N resursov iz skladišča z imenom storage name

LEAVE ␣ storage name, M - vračanje M resursov v skladišče z imenom storage name.

Primer 2: isto samo da delajo 3 brivci, prihajajo 6±4 min stranke se ne razlikujejo med brivci

```
         DA SO TRIJE BRIVCI
BRIVEC   STORAGE,3     } deklaracija

GENERATE 6,4         )
QUEUE     VRSTA      |
ENTER     BRIVEC,1   |
DEPART    VRSTA      }  B1
ADVANCE   13,2       |
LEAVE     BRIVEC,1   |
TERMINATE            )

                     }  B2

START 1                } kontrolni ukaz
```

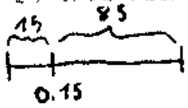
DELAVNICA

TRANSFER - skok transakcije

TRANSFER ␣ ① labela - brezpogojni skok na vrstico z imenom labela

TRANSFER ␣ .15, labela1, labela2 - 15% transakcij, ~~[illegible]~~ skoči na lokacijo labela2, ostalih 85% pa na labela1

0.15



TRANSFER ␣ BOTH, labela1, labela2 - transakcijo se pošlje na stavek labela1, če je ta zaseden se ~~[illegible]~~ transakcijo pošlje na stavek z labelo2, če pa je tudi ta zaseden, se izmenično preverja, kateri stavek ostane prost in tja se usmeri transakcija

PRIMER: Tovarna TV, kjer se vrši končna kontrola. Kontrolo neodvisno vršita 2 delavca. V primeru napake gre TV na servis, kjer dela 1 delavec. Po servisu gre TV zopet v kontrolo. TV-ji prihajajo v kontrolo: 5'5 ± 2 min. Kontrola TV-ja: 9 ± 3 min. 85% TV-jev je brezhibnih, 15% jih gre na servis. Servis traja 30 ± 10 min. Simuliraj 8 urni delavnik.

B1

```
TESTER      STORAGE,2
GENERATE    55,20
(A QUEUE    KONTROLA
ENTER       TESTER,1
DEPART      KONTROLA
ADVANCE     90,30
LEAVE       TESTER,1
TRANSFER    .15,,SERVIS
TERMINATE
```

B2

```
SERVIS QUEUE     SERVISIRANJE
       SEIZE     SERVISER
       DEPART    SERVISIRANJE
       ADVANCE   300,100
       RELEASE   SERVISER
       TRANSFER  ,DELAVNICA
```

B3

```
GENERATE 4800
TERMINATE 1
```

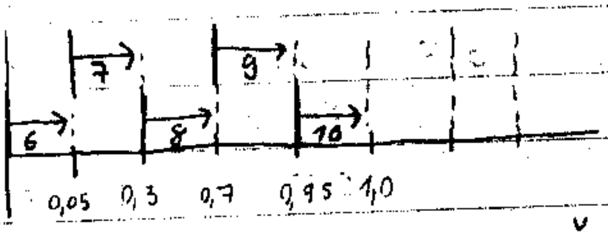
```
START 1
```

Tovarna ima 4 delavce, ki med seboj neodvisno sestavljajo izdelke. Izdelki gredo po sestavljanju v peč (v tovarni je le ena peč). V peči se lahko peče največ 1 izdelek. Sestava: 30 ± 5 min. Simuliraj 40 delovnih ur. Pečenje traja.

trajanje	relativne frekvence	kumulativne frekvence ∅
6	0,05	0,05
7	0,25	0,3
8	0,4	0,7
9	0,25	0,95
10	0,05	1,0

```
      D  [ PECENJE     FUNCTION  RN1,D5
         [ .05,6/.3,7/.7,8/.95,9/1.0,10
ZACETEK  [ GENERATE    ,,,4
           ADVANCE     30,5
      B1   SEIZE       PEC
           ADVANCE     FN$PECENJE
```

pokliči f. z imenom pečenje, ki naj mi vrne neko vrednost



RN1 → X

0 ≤ X < 1

```
         [ TRANSFER    ,ZAČETEK

      B2 [ GENERATE    2400
         [ TERMINATE   1

           START       1
```

Na uro v banko vstopi 240 ljudi. Imamo 8 ekvivalentnih okenc. Za
je temi okenci imamo 1 čakalno vrsto. Simuliraj 5 urni delovnik, če
stranke opravljajo 5 vrst opravil po tabeli.

vrsta posla	trajanje posla	relativne frekvence	kumulativa
1	45 s	0,1	0,1
2	75 s	0,19	0,29
3	100 s	0,32	0,61
4	150 s	0,24	0,85
5	300 s	0,15	1,0

Transakcijski parametri

$\{P_1, P_2, \ldots, P_N\}$ $t=0 : \forall P_i = 0$

ASSIGN i, j ...... $P_i := j$

ASSIGN i+, j ...... $P_i = P_i + j$

{če je $P_i := 0 \Rightarrow P_i = 1$}

M1 ... življenski čas

240:60 = 4 lju

1 človek/15 s

12 s

```
  D  ⎡ OKNO          STORAGE 8
     ⎢ POSEL         FUNCTION RN1, D5.
     ⎣ .1,45/.29,75/.61,100/.85,150/1.0,300
     ⎡ GENERATE      15, 3
     ⎢ QUEUE         VRSTA
 B1  ⎢ ENTER         OKNO, 1
     ⎢ DEPART        VRSTA
     ⎢ ASSIGN        1, FN$POSEL ...... P1 = trajanje
     ⎢ ADVANCE       P1
     ⎣ TERMINATE

 B2  ⎡ GENERATE      18000
     ⎣ TERMINATE     1

       START         1
```

✓σ

SELECT ⎵O⎵ A,B,C,D,E,F ; O - operator

E =

G >

L <

GE ≥

LE ≤

MIN

MAX

A - transakcijski parameter, v katerega se izpiše rezultat izbire

B,C - spodnjo in zgornjo mejo izbirnega področja

F - je labela, če do izbire ni ~~[illegible]~~ / potrebe

E - sistemska etiketa : - vrsta

- resurs

:

- spremenljivka

D - vrednost, ki je operator E

PRIMER: Imamo 8 carinskih vrst. Stranki je vseeno za katerim okencem opravlja storitev.

D { POSEL FUNCTION RN1,D5

0,1, 4,5 / ....

GENERATE 15,3

ASSIGN 1, FN$POSEL ↗ skočimo, če nismo našli praznega okenca

SELECT E 2, 1, 8, Ø, F, SKOK → facility (busy = 1, free = 0)

P2 - indeks praznega ~~vrste~~ okenca

kandidati za opazovanje

SKOK 1 QUEUE P2 V ← ime vrste

SEIZE P2 R ← ime resursa

DEPART P2
ADVANCE P1] B1
RELEASE P2
TERMINATE

tekoča vsebina vrste

SKOK SELECT MIN 2,1,8,,Q] B2
TRANSFER ,SKOK1.

GENERATE 1800] B3
TERMINATE 1

operand E

TEST E A B .... dokler A in B nista enaka, transakcija
↳(=, >, <, ≥, ≤, ...) ne zapusti stavka

TEST E A B C. .... če sta A in B različna, transakcija
skoči na labelo C, drugače pa
zapusti TEST stavek

SAVE VALUE prirejanje spremenljivke

SAVEVALUE A,B ... A := B

SAVEVALUE A+,B ... A := A+B

"B_i" → TABULATE A ... generira tabelo A

Potrebna predhodna deklaracija

D: A TABLE X$B, 0, 10, 3 ← # razredov

vrednost B, ki se tabelira — spodnja meja 1. razreda — interval razredi

1	0..9
2	10..19
3	20..29

* FN$ Funkcija (Funcname) .. klic evaluacije vrednosti funkcije

* X$ Var_name ... branje trenutne vrednosti spremenljivke var_name

* V$ Var_name ... klic evaluacije vrednosti spremenljivke

SAVEVALUE : Var_Name

* spremenljivke: read : X$ Var_Name
izračun : V$ Var_Name

Skladišče dnevno izdaja 10 ± 2 kosa. Ko zaloga pade pod ROP (reader point) se naroči nova pošiljka, ki rabi 8 dni. Bude ROQ (Reader quantity) kosov. Tabeliraj dnevne zaloge. Simuliraj 1000 dni.

```
          D   [ PORABA      FUNCTION     RN1, D5
              [ 0,2 ,8/0.4, 9 /0.6, 10/ 0.8, 11, /1.0, 12
              [ SKLADIŠČE   TABLE        X$ZALOGA, 0, 10, 12
              [ IZGUBA      VARIABLE     P1- X$ZALOGA

              [ GENERATE                 1
              [ ASSIGN                   1, FN$PORABA
          B1  [ TEST   GE                X$ZALOGA, P1, PROBLEM
              [ SAVEVALUE                ZALOGA -, P1
SKOK 1        [ TABULATE                 SKLADIŠČE
              [ TERMINATE                1

PROBLEM       [ ASSIGN                   2, V$IZGUBA
          B2  [ SAVEVALUE                ZALOGA, 0
              [ TRANSFER                 , SKOK 1

              [ GENERATE                 ,,,, 1
              [ TEST   L[illegible]      X$ZALOGA, X$ROP
          B3  [ ADVANCE                  8
              [ SAVEVALUE                ZALOGA +, X$ROQ
              [ TRANSFER                 , SKOK 2

> INITIAL   X$ROP, 80      poznamo sprem. 80 v initial ROP
> INITIAL   X$ROQ, 100
> INITIAL   X$ZALOGA, 100
> START     1000
```

SEMINARSKE

GPSSPC.EXE ~~GPSSREPT.EXE~~

- vrstični editor
- zagon simulacije

ukazi:
> @VAJA.GPS Open file
> SAVE VAJA.GPS
> DISPLAY Izpis izvorne kode
> END Izhod iz aplikacije
> DELETE 100 Izbriše vrstico 100
> DELETE 100-200
> EDIT 100

GPSSREPT.EXE

branja in tiskanja reporta Report.GPS

POROČILA

① Osnovne karakt. ⟶ stat_time ... 0
② Facilitice end_time ... 1200
③ QUEUE facilities ... 3
④ Tabele storages ... 0
blocks ... 32

UKAZ

$\underbrace{\text{addresa vrstice}}$ $\underbrace{\text{labela}}$ $\underbrace{\text{UKAZ}}$ $\underbrace{\text{PARAMETRI}}$

1 GENERATE
2

Entry Count
17
↑
št. trans., ki so vstopile v ta stavek

Current Count
0
↑
št. transakcij, ki so obstale v tem stavku po koncu simulacije.

② Facilities → SERVISER

Entries
12
kolikokrat je bil resours zaseden

Avaible
1/0
1 = prost po koncu transakcije

Utilization
0'44
Procent zasedenosti resoursa skozi simulacijo.

Owner
7
↑
indeks transakcije, ki po transakciji zadržuje resours

Ave. time
21
povp. čas zasedenosti resoursa s strani ene transakcije

AVE.C.
UTIL.

③ Queue → SERVIS

① Max	② CONT.	③ ENTRIES
3	0	12
max. št. čakajočih	št. čakajočih po koncu simulacije	št. vhodov v vrsto skozi simulacijo

④ ENTRIES (0)	⑤ Ave. Cont.	⑥ Ave. Time
7	0'12	10.641
št. transakcij ki vstopijo v prazno vrsto	povp. čas čakanja v vrsti glede na čas transakcije v sistemu	povp. čas čakanja v vrsti

⑦ Ave. (-0)
22.6
Povp. čas čakanja
v vrsti brez
transakcij, ki so
vstopile v prazno vrsto

④. Tabele → SKLADIŠČE

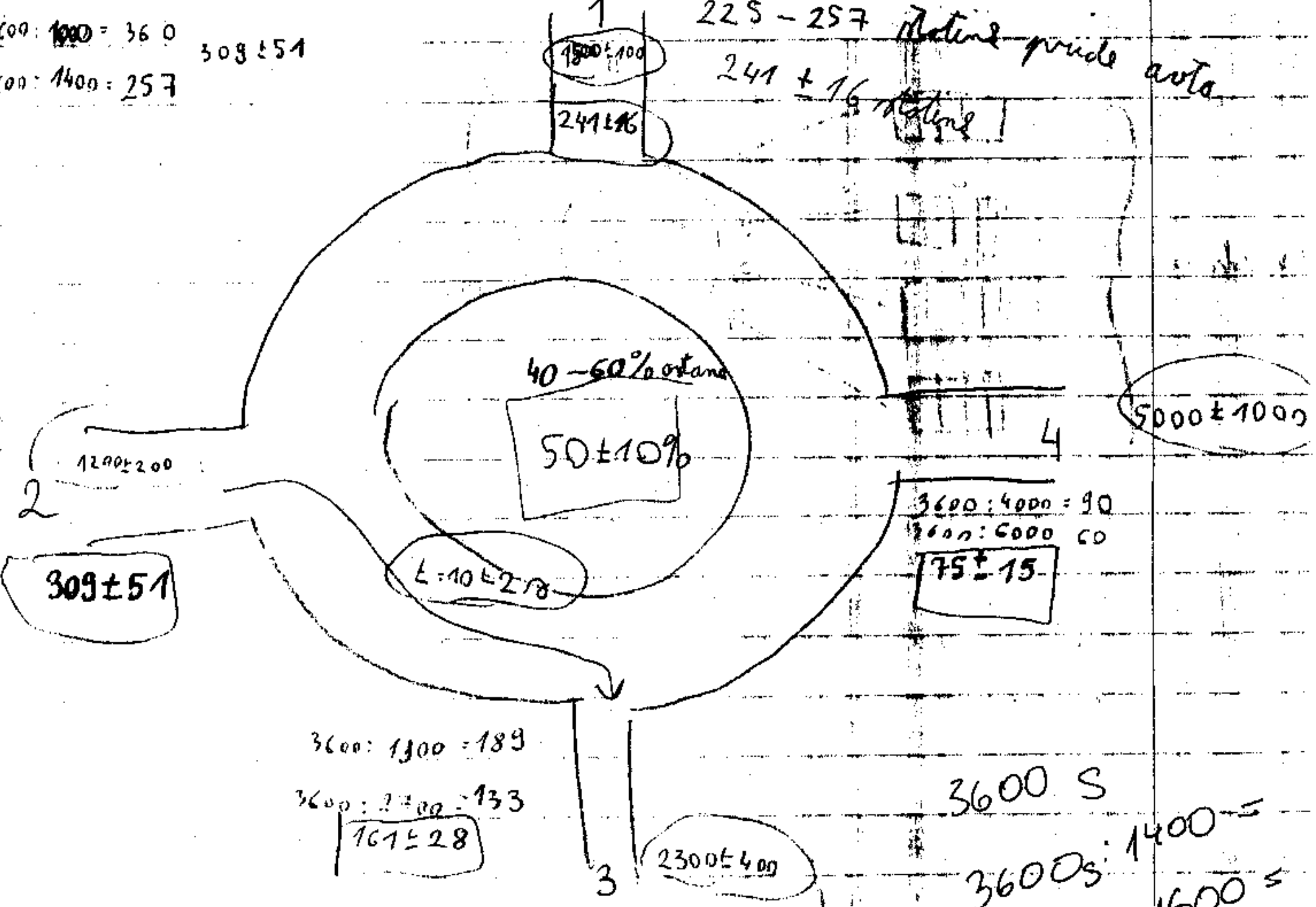
MEAN	STD. DEV		freq.
105	20.7	0..9	6
povp. vrednost		10..19	27
		20..29	967

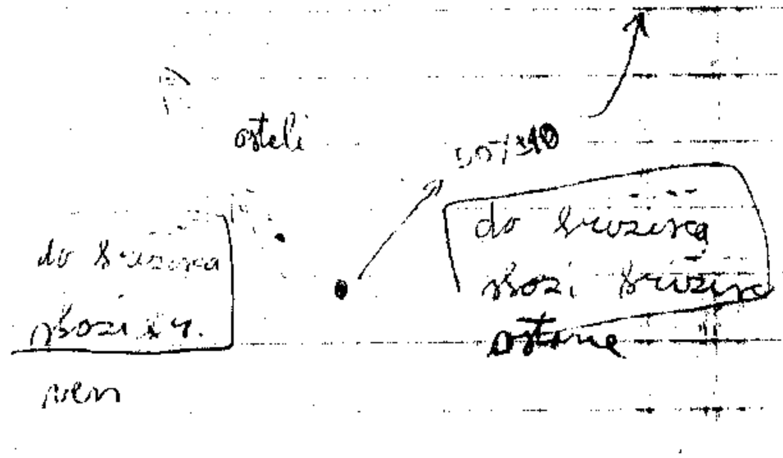
SEMINARSKA 14. NALOGA

3600 : ~~1000~~ = 360

3600 : 1400 = 257

309±51





3600 S

3600s : 1400 =

3600s : 1600 =

3600 S

257 stotink

225 stotink

241 stotink ±16 stot

10±2

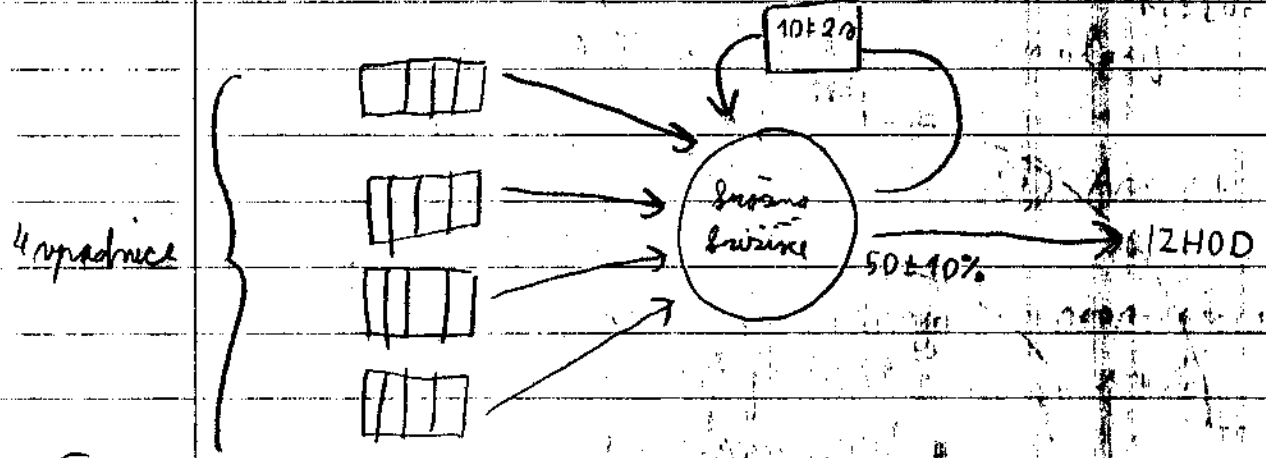
8 r + 40% - 11,2 r

40%

12 r + 60% - 15,2 r

60%

SEMINARSKA



2. naloge na pulu

Petrijeve mreže (PM)

T akcija ▬ (tranzition - prehajanje)
P pogoji O (place)
žetoni • len žeton v pogoju pomeni da je pogoj 1x izpolnjen

Pogoj: Imam 7 sit za sladoled
(•) 7 sit
(••) 14 sit
(••) 17 sit

Petrijeva mreža je četvorček

$$C = (P, T, \overset{\text{vhod}}{I}, \overset{\text{izhod}}{O})$$

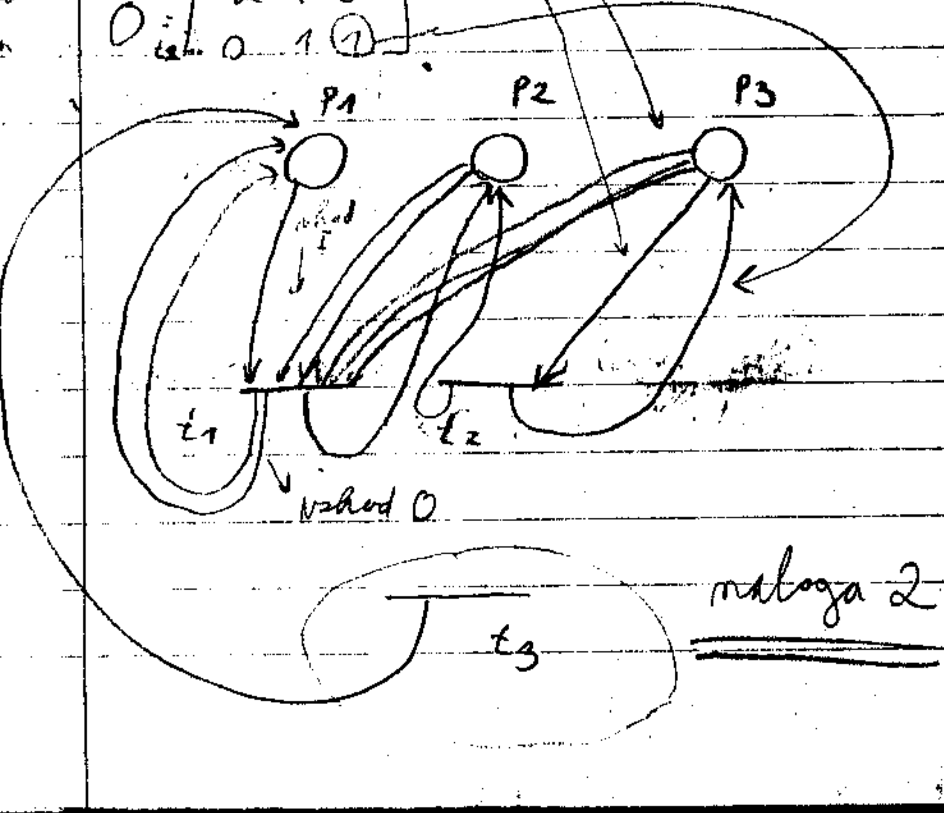
matrike

1. Nariši graf PM iza matriki

vhodna matrika $$\underline{I} = \begin{array}{c|ccc} & & & \\ t_1 & 1 & 2 & \textcircled{3} \\ t_2 & 0 & 0 & \textcircled{1} \\ & P_1 & P_2 & P_3 \rightarrow \text{pogoj} \end{array}$$

izhodna matrika $$O = \begin{array}{c|ccc} t_1 & 2 & 1 & 0 \\ t_2 & 0 & 1 & \textcircled{1} \end{array}$$

obe matriki glasita na akcije



2. PM iz naloge 1 dodaj enostavni (enojni) vhod v P1 iz t3. PM nato zapiši z matrikami

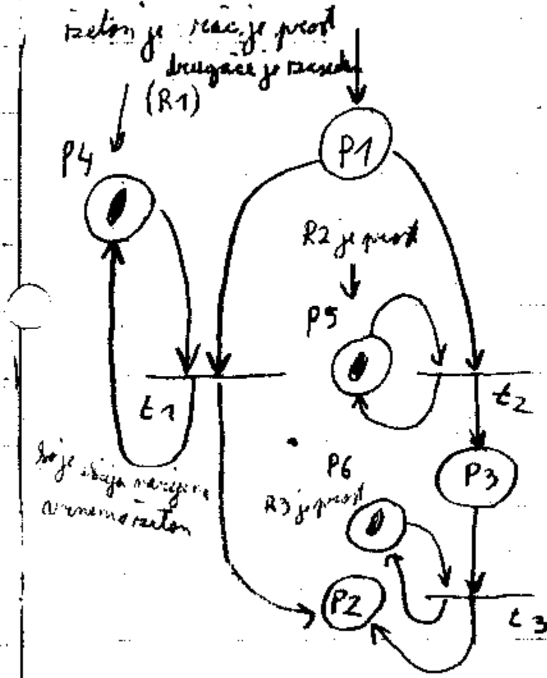
ofen.

$$I = \begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array}\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{array}{l} \} \text{ enako} \\ \\ \rightarrow \text{v } t_3 \text{ nič ne vstopa} \end{array}$$

$$O = \begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array}\begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{array}{l} \} \text{ enako} \\ \\ \rightarrow \text{iz } t_3 \text{ v } P1 \end{array}$$

!!!

3. V rač. sistem vstopajo zahteve, ki so lahko obdelane na R1 ali pa zaporedno na R2, R3. Nariši Petrijev graf in obrazloži pomen posameznih akcij in pogojev



P1 ... v sistem vstopi zahteva;
t1 zahtevo obdeluje R1;
P2 ... zahteva je obdelana; (koliko žetonov v P2, toliko jih je sistem obdelal)
t2 ... zahtevo obdeluje R2 (računalnik 2);
P3 ... R2 je opravil delo na zahtevi
t3 ... zahtevo obdeluje R3.
P4 ... pogoj R1 je prost

// (iz akcije v akcijo ali pogoja v pogoj) NI - V) REDU +

√ iz akcije v pogoj ali pogoj v akcijo

P4 ... R1 je prost
P5 ... R2 je prost
P6 ... R3 je prost

MOČ PM da se determinizem izvaja (iščemo luknje v sistemu)

A

4.

Zapisati je treba začetno označitev.
(to je razporeditev žetonov v pogojih)

P2 je 0, da vidimo točno
↓ katero smo prišli

$$(\overset{\uparrow}{0}, \overset{\downarrow}{0}, \underset{P3}{\overset{}{0}}, 1, 1, 1)$$

v P1 damo 0 - ker mi ne odločamo, ampak nekdo drug

Pogoj za proženje akcije t1, oba žetona P1 in P4.
Eden gre v P2, drugi se vrne v P4.

3. Proženje akcij v PM

- Pogoj da akcija steče so izpolnjeni vhodni pogoji ≈ to velja takrat, kadar je število žetonov v vseh vhodnih pogojev večje ali enako številu kratnosti posameznih povezav.
- Kdaj t_j steče:

akcija t_j

$$\underset{\substack{\downarrow \\ \text{trenutna} \\ \text{označitev } (\sigma)}}{\sigma} \geq e[j] \cdot I$$

$e[j] = (0, 0, \ldots, \overset{\substack{j\text{-to mesto je 1} \\ \downarrow}}{1}, 0)$

- Kaj se zgodi, če t_j steče?

$$\sigma' = \sigma + e[j] \cdot (O - I)$$

4. Imamo PM podano z

$$I = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

$$O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Kam nas pripelje sproženje akcije ⓣ₂ ob označitvi $(1,1,1,1)$?

za 2. mesto ①

$\sigma \geq e[j] \cdot I$

$$(1,1,1,1) \geq (0,1,0) \cdot \underline{I} = (0,0,0,1)$$

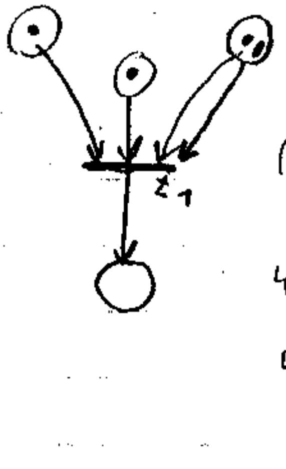
$$\sigma' = (1,1,1,1) + (0,1,0) \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

0 žetonov v 4. pogoju

$$= (1,1,1,1) + (0,2,1,-1) = (1,3,2,0)$$

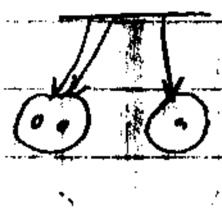
1 žeton v prvem pogoju

3 žetoni v 2. pogoju



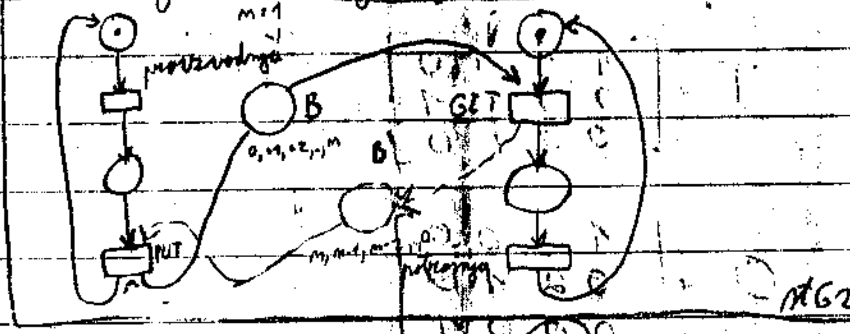
da povezava pasivi, se mora pa njej pripeljat 1 žeton (vx4, da gre naprej 1 žeton)

Če ima akcija uspeh, in ima 2 vzhodne povezave gredo 3 žetoni ven.

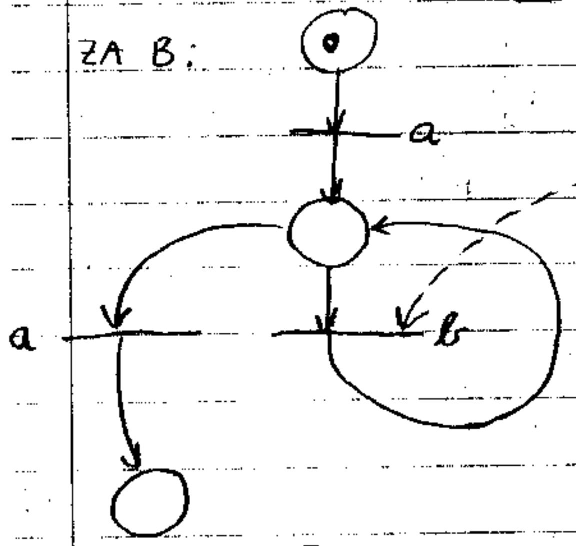


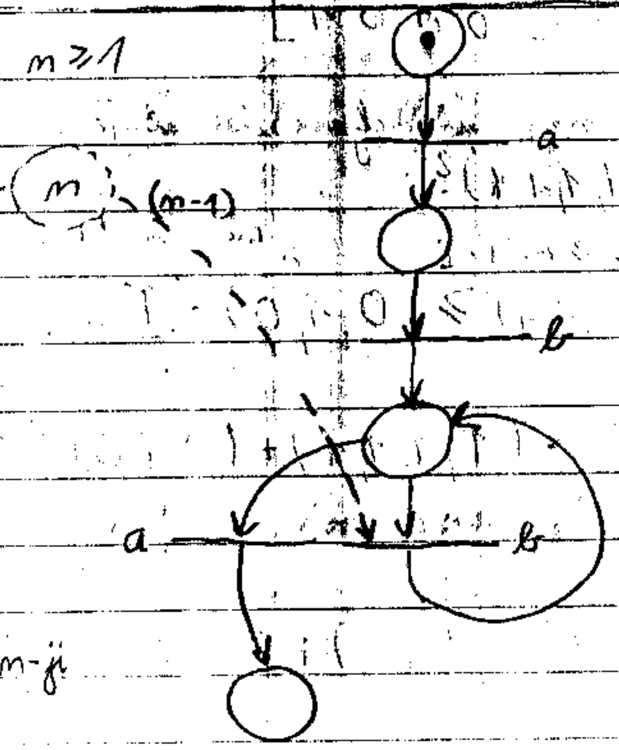
!!! IZPIT ① Konstruiraj PM za generiranje jezika "ab^{n}a"

n≥1
aba
abba
abbba
ab...ba



ZA B:



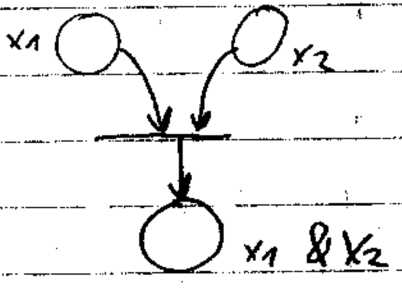


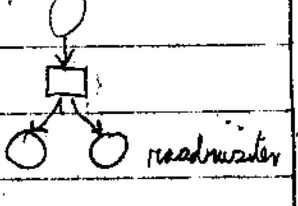
(n) s tem imamo nadzor nad n-ji

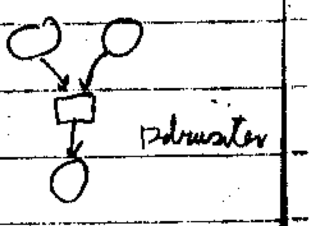
FOR
znati
WHILE
IF

② Osnovne logične operatorje AND, OR, XOR ⇒ programski konstrukt semafor realiziraj s Petrijevo mrežo.

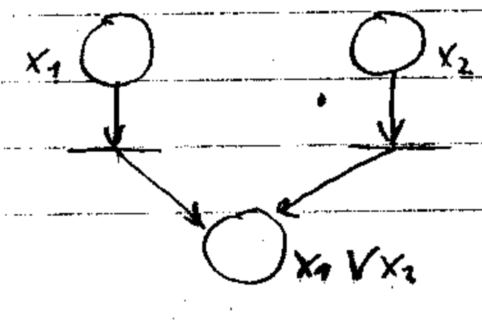
a) x_1 & x_2

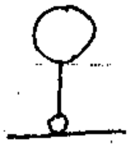






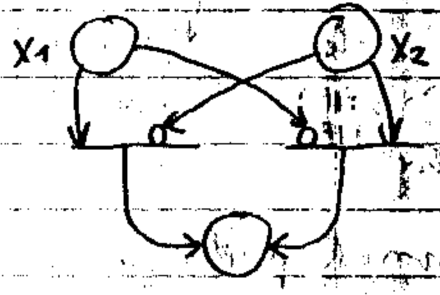
b) $x_1 \vee x_2$



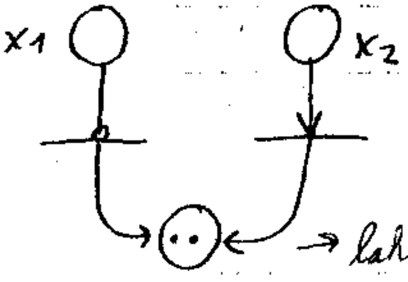


- negiran ; al pravilo je izpolnjen kadar ni res
akcija je sprožena, ko pogoj ne velja!!!

c) XOR x_1 XOR x_2

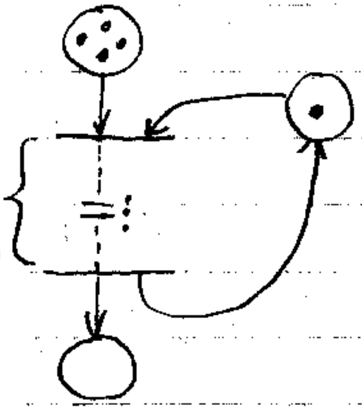


d) $X_1 \Rightarrow X_2 \sim \bar{X}_1 \vee X_2$



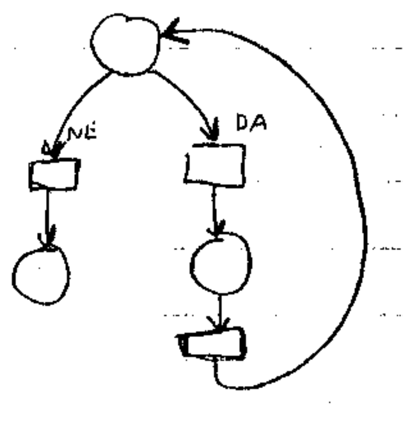
→ lahko pridemo do dveh žetonov, ampak nas to ne moti.

e) SEMAFOR

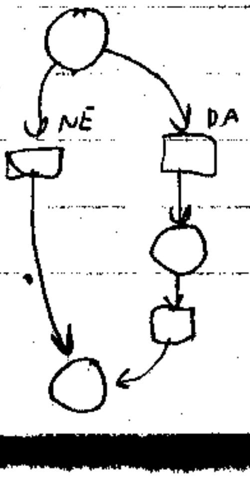


v vsem naj bi bil trenutno le en žeton

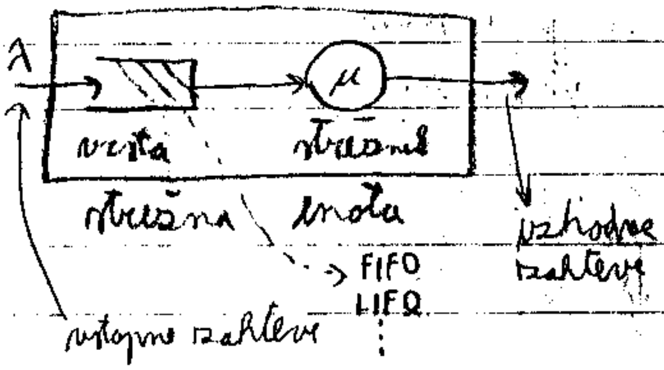
WHILE



IF



ZADNJA URA



λ .. intenzivnost prihajanja sistema
μ .. intenzivnost procesiranja zahtev
ρ .. uporabnostni faktor strežne enote (SE) $\boxed{\rho = \frac{\lambda}{\mu} = \lambda \cdot \bar{x}}$
→ povprečni čas procesiranja ene zahteve

$\boxed{N = \lambda \cdot T}$ Littlevo pravilo $\boxed{\mu = \frac{1}{\bar{x}}}$

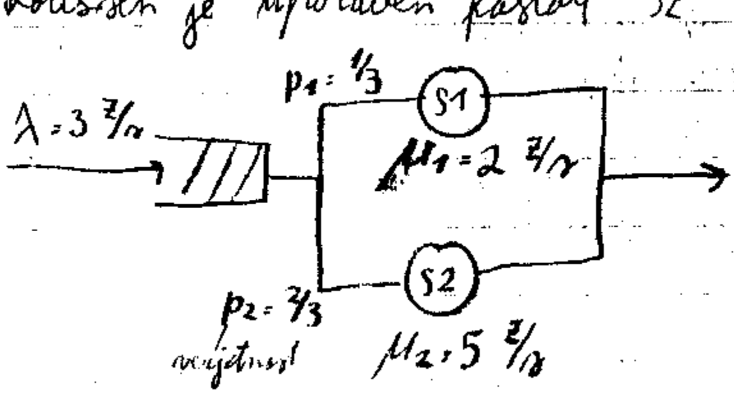
→ povprečno št. zahtev v sistemu
→ povprečni čas bivanja zahteve v sistemu

? $\boxed{T = W \cdot \bar{x}}$

→ čas čakanja v vrsti

$T = W + \bar{x}$

1. Kolikšen je uporaben faktor SE na diski

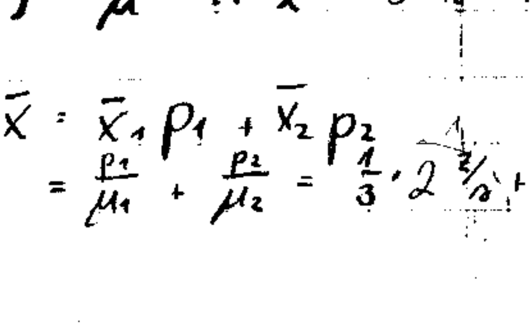


$\boxed{\bar{X}}$

$$\rho = \frac{\lambda}{\mu} = \lambda \cdot \bar{x} = 3\,z/s \cdot \frac{3\,s}{10\,z} = \underline{\underline{\frac{9}{10}}}$$

$$\bar{x} = \bar{x}_1 p_1 + \bar{x}_2 p_2 = \frac{p_1}{\mu_1} + \frac{p_2}{\mu_2} = \frac{1}{3}\cdot \overset{1}{2}\,z/s + \frac{2}{3}\cdot \overset{1}{5}\,z/s = \frac{5+4}{30}\frac{s}{z} = \frac{9}{30}\,s/z = \boxed{\frac{3\,s}{10\,z}}$$

2. Avtomat dvigala, ki vozi med pritličjem in prvim nadstropjem v diskretnih časovnih korakih preverja ali je prišlo do zahteve za vožnjo. Kolikšna je verjetnost, da smo po 4. koraku v 1. nadstropju, če smo bili na začetku v pritličju in velja naslednja shema.



← verjetnost da v enem koraku ostanemo v 1. nadstropju

iz 0 v 1

$$M = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 3/4 & 1/4 \\ 1 & 1/2 & 1/2 \end{array} \qquad M^4 = \begin{bmatrix} & \bigcirc \\ & \end{bmatrix}$$

$$\underline{\underline{p_{2\to1}(4) = \frac{1}{256}\cdot 85}}$$

upoštevamo →

$$M^2 = \frac{1}{4}\cdot\frac{1}{4}\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 11 & 5 \\ 10 & 6 \end{bmatrix}$$

$$M^4 = M^2 \cdot M^2 = \frac{1}{16}\cdot\frac{1}{16}\begin{bmatrix} 11 & 5 \\ 10 & 6 \end{bmatrix}\begin{bmatrix} 11 & 5 \\ 10 & 6 \end{bmatrix} = \frac{1}{256}\begin{bmatrix} 171 & \textcircled{85} \\ 170 & 86 \end{bmatrix}$$

3, V sistemu M/M/1 se v povprečju nahaja 8 zahtev. Kolikšna je verjetnost praznega sistema?

$$M/M/m/m/k$$

eksponentno porajanje zahtev na vhodu in eksponentno procesiranje

št. strežnikov

dolžina vrste

velikost populacije

$$\boxed{N = \ldots\ldots = \frac{\rho}{1-\rho}} \qquad \boxed{p_0 = 1 - \rho}$$

$N = \rho =$

$$8 = \frac{\rho}{1-\rho}$$

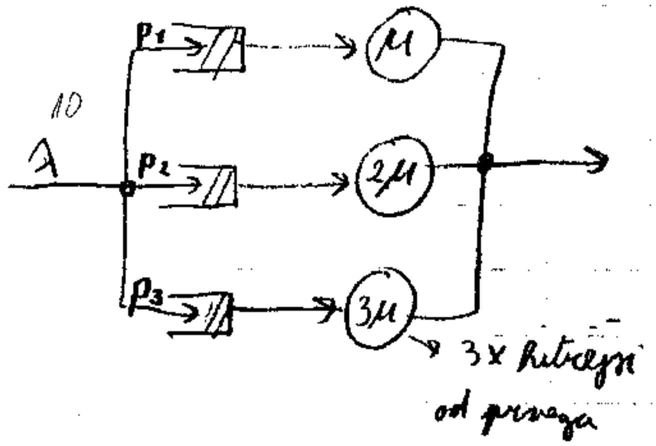
$$8 - 8\rho = \rho$$

$$8 = 9\rho$$

$$\rho = \cancel{\ } = \frac{8}{9}$$

$$\boxed{p_0 = 1 - \frac{8}{9} = \frac{1}{9}}$$

4. Kolikšen je čas prebivanja zahtev v sistemu s tremi enostavnimi strežnimi enotami, če so $p_1=0'1$; $p_2=0'5$ in $p_3=0'4$.
Vsaka strežna enota je M/M/1; intenzivnost vhoda pa je 10 z/s



$T = P \; \& \; \frac{1}{M}$

$$T = T_1 p_1 + T_2 p_2 + T_3 p_3$$

$$\boxed{T = \frac{\bar{N}}{\lambda} = \dots = \frac{1}{\mu - \lambda}}$$ enačica Littnovega pravila

$$T_s = \frac{1}{10(\mu-1)} + \frac{5}{10(2\mu-5)} + \frac{4}{10(3\mu-4)}$$