Petrijeve mreže

Carl Adam Petri, doktorska disertacija 1962. godine

$$PM = (P, T, I, O, \Phi, m_0)$$

$P = \{p_1, p_2, ..., p_m\}$ - skup svih **mjesta** (*places*) u PM,

$T = \{t_1, t_2, ..., t_n\}$ - skup svih **prijelaza** (*transitions*) u PM,

I : T=>P - ulazna funkcija, preslikavanje sa skupa prijelaza u skup mjesta (matrica veza od mjesta prema prijelazima), **ulazna matrica događaja**,

O : T=>P - izlazna funkcija, preslikavanje sa skupa prijelaza u skup mjesta (matrica veza od prijelaza prema mjestima), **izlazna matrica događaja**,

Φ : (I, O) => {1, 2, ...} - težinska funkcija (**matrica svih težinskih koeficijenata u PM**),

m_0 : P => {0, 1, 2, ...} - **početno stanje** (*initial marking*) vektora stanja m.

Graf Petrijeve mreže => bipartitni multigraf

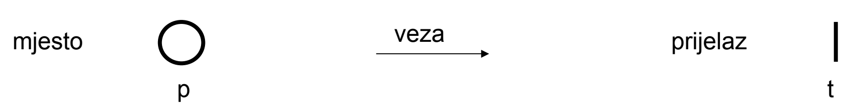
$$G = (V, A)$$

bipartitni – dvije vrste čvorova,

multigraf – moguće više veza između dva čvora.

V - skup **čvorova** (*nodes*), $V = P \cup T$,

A - skup **usmjerenih veza** (*directed arcs*), koje povezuju elemente skupa V.



Usmjerene veze povezuju različite vrste čvorova.

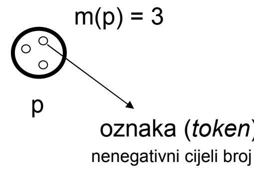
obična Petrijeva mreža =>

$$\forall a(p,t)\ :\ a(p,t) = 1$$
$$\forall a(t,p)\ :\ a(t,p) = 1$$

Vektor stanja (*marking vector)* Petrijeve mreže

$$m:\ P \to N_0$$

stanje mjesta p => m(p)



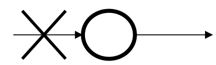
skup ulaznih mjesta prijelaza t

$$\bullet\, t = \{ p \in P \mid a(p,t) > 0 \}$$

skup izlaznih mjesta prijelaza *t*

$$t \,\bullet = \{ p \in P \mid a(t,p) > 0 \}$$

izvor (*source*)



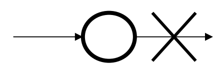
skup ulaznih prijelaza mjesta p

$$\bullet\, p = \{ t \in T \mid a(t,p) > 0 \}$$

skup izlaznih prijelaza mjesta p

$$p \,\bullet = \{ t \in T \mid a(p,t) > 0 \}$$

ponor (*sink*)



Primjer: Petrijeva mreža

$P = \{p_1, p_2, p_3, p_4, p_5\}$, $T = \{t_1, t_2\}$,

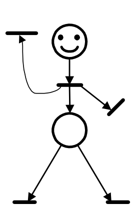
$V = \{p_1, p_2, p_3, p_4, p_5, t_1, t_2\}$,

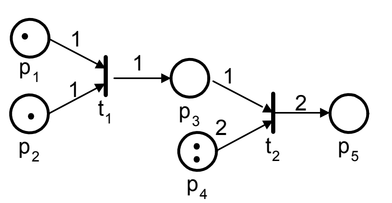
$a(p_1,t_1)=1$, $a(p_2,t_1)=1$,
$a(p_3,t_2)=1$, $a(p_4,t_2)=2$,
$a(t_1,p_3)=1$, $a(t_2,p_5)=2$.

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

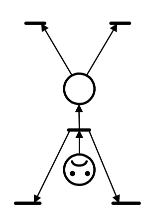
$$\Phi = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix},$$

$m = [m(p_1)\ \ m(p_2)\ \ m(p_3)\ \ m(p_4)\ \ m(p_5)]^T$, $m_0 = [1\ 1\ 0\ 2\ 0]^T$.



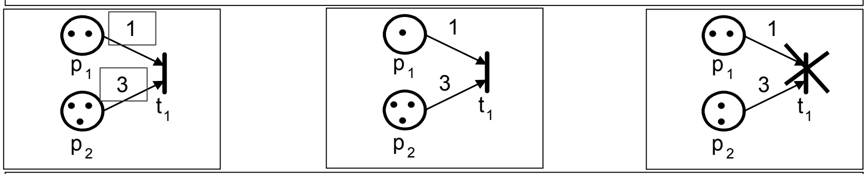


$\bullet t_1 = \{p_1, p_2\}$ $\bullet p_5 = \{t_2\}$

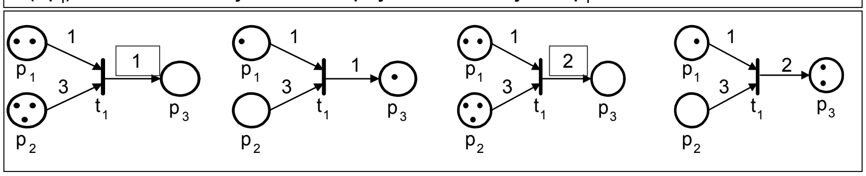


Pravilo okidanja Petrijeve mreže

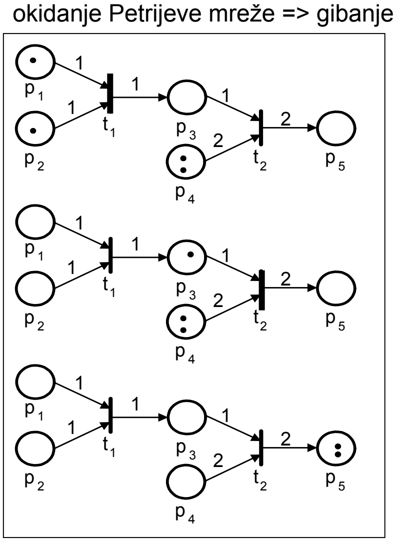
Prijelaz t je **omogućen** (*enabled*) ako je svako ulazno mjesto p označeno s barem a(p,t) oznaka, gdje je a(p,t) težinska funkcija veze iz mjesta p u prijelaz t.



Okidanje omogućenog prijelaza t oduzima $a(p_u,t)$ oznaka svakom ulaznom mjestu p_u prijelaza t i dodaje $a(t,p_i)$ oznaka svakom izlaznom mjestu p_i, gdje je $a(t,p_i)$ težinska funkcija veze od prijelaza t do mjesta p_i.



okidanje Petrijeve mreže => gibanje oznaka => promjena vektora stanja



$m_0 = [1\ 1\ 0\ 2\ 0]^T$

$m_1 = [0\ 0\ 1\ 2\ 0]^T$

$m_2 = [0\ 0\ 0\ 0\ 2]^T$

Jednadžba prijelaza stanja Petrijeve mreže

$$m_{t2} = m_{t1} + W^T \cdot \tau$$

m_{t1} - vektor stanja u trenutku t_1,

m_{t2} - vektor stanja u trenutku t_2,

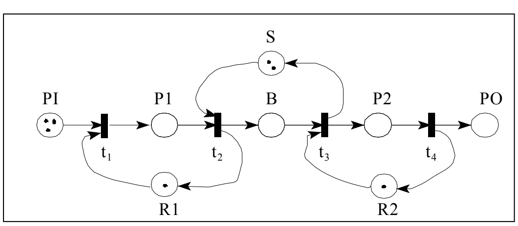
τ - vektor koji sadrži elemente τ_i (broj propaljivanja prijelaza *i* u intervalu od trenutka t_1 do trenutka t_2).

$W = O - I$ - **matrica događaja** (*incidence matrix*)

Zanima nas vrijednost vektora stanja nakon svakog pojedinog propaljivanja bilo kojeg prijelaza:

$$m_k = m_{k-1} + W^T \cdot \tau_k$$

Primjer: jednadžba prijelaza stanja Petrijeve mreže



P={PI, P1, B, P2, R1, S, R2, PO}.

T={t_1, t_2, t_3, t_4}.

m_0 = [3 0 0 0 1 2 1 0]^T

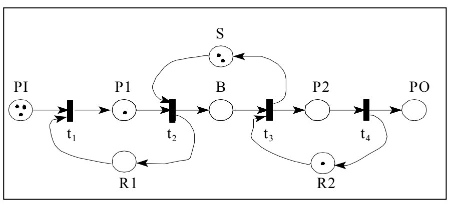
τ_1 = [1 0 0 0]^T

m = [m(PI) m(P1) m(B) m(P2) m(R1) m(S) m(R2) m(PO)]^T.

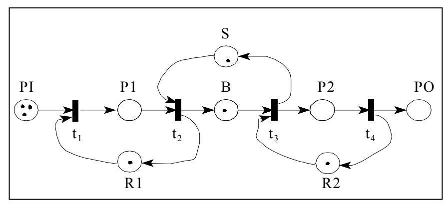
$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad O = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 &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

$$W = \begin{bmatrix} -1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$m_1 = \begin{bmatrix}2\\1\\0\\0\\0\\2\\1\\0\end{bmatrix} = \begin{bmatrix}3\\0\\0\\0\\1\\2\\1\\0\end{bmatrix} + \begin{bmatrix}-1&0&0&0\\1&-1&0&0\\0&1&-1&0\\0&0&1&-1\\-1&1&0&0\\0&-1&1&0\\0&0&-1&1\\0&0&0&1\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \qquad m_2 = \begin{bmatrix}2\\0\\1\\0\\1\\1\\1\\0\end{bmatrix} = \begin{bmatrix}2\\1\\0\\0\\0\\2\\1\\0\end{bmatrix} + \begin{bmatrix}-1&0&0&0\\1&-1&0&0\\0&1&-1&0\\0&0&1&-1\\-1&1&0&0\\0&-1&1&0\\0&0&-1&1\\0&0&0&1\end{bmatrix}\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$



$\tau_2 = [\,0\ 1\ 0\ 0]^T$



$\tau_3 = [\,1\ 0\ 1\ 0]^T$

Osnovna svojstva Petrijevih mreža

Dohvatljivost – m_k je dohvatljiv iz m_j: $m_k \in \Re(m_j)$

postoji niz okidanja prijelaza takav da vektor stanja prelazi iz m_j u m_k

Ograničenost – $\forall p : m(p) \le M$

Živost – moguće je okinuti sve prijelaze u mreži (vrlo strog kriterij; zaglavljenje <=> živost)

Reverzibilnost – $\forall m, m \in \Re(m_0) : m_0 \in \Re(m)$ cikličko ponašanje sustava

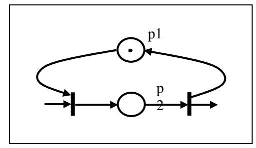
Jednoznačnost (determiniranost) – t_1 i t_2 omogućeni => okidanje t_1 ne onemogućava okidanje t_2 (konflikt)

Strukturna svojstva Petrijevih mreža **(ne ovise o vektoru stanja)**

P-invarianta je binarni P-vektor (vektor mjesta) za koji vrijedi:

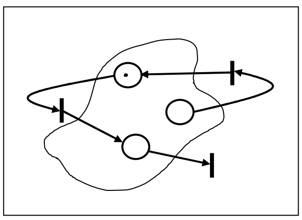
$$W \cdot p = 0 \longrightarrow p^T m_k = p^T m_0$$

Suma oznaka u mjestima invarianti je konstantna.



Sifon - skup mjesta za koji vrijedi

$$\bullet S \subset S \bullet$$

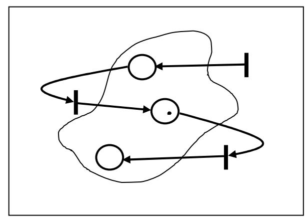


Svaki prijelaz koji ima izlazno mjesto u sifonu ima i ulazno mjesto u sifonu.

Ako sifon izgubi oznake, nikada više neće moći primiti oznaku.

Zamka – skup mjesta za koje vrijedi

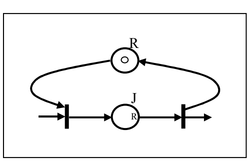
$$Q \bullet \subset \bullet Q$$



Svaki prijelaz koji ima ulazno mjesto u zamci ima i izlazno mjesto u zamci.

Oznaka koja uđe u zamku ne može je nikada napustiti.

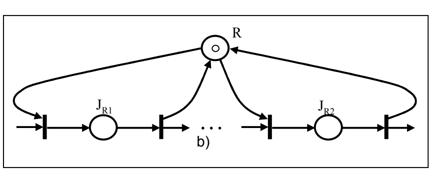
Petrijeva mreža ⇔ fleksibilni proizvodni sustav



jednoradni resurs: m(R) = 1 => resurs je slobodan

$m(J_R) = 1$ => resurs obavlja posao J_R

višeradni resurs: m(R) = 1 => resurs je slobodan



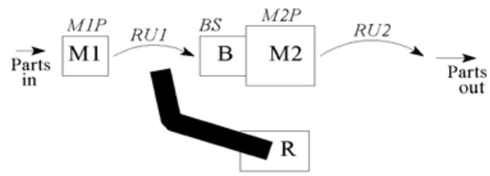
$m(J_{R1}) = 1$ => resurs obavlja posao J_{R1}

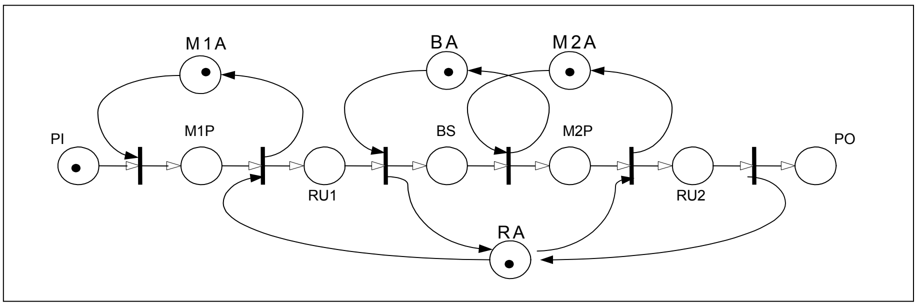
$m(J_{R2}) = 1$ => resurs obavlja posao J_{R2}

$$r_i \cup J(r_i) = p_{inv}$$

Resurs sa skupom poslova koje obavlja, čini p-invariantu.

Primjer: fleksibilni proizvodni sustav opisan Petrijevom mrežom

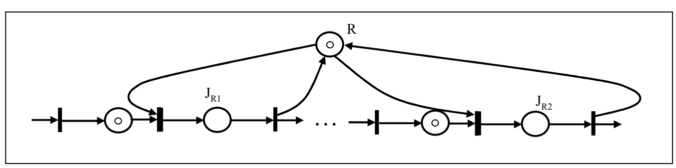




Problemi ?

Konflikt

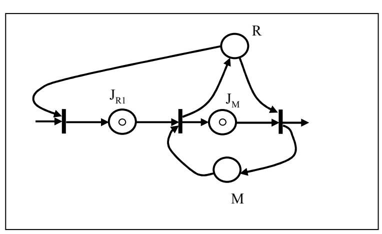
Istovremeno je omogućeno obavljanje nekoliko operacija višeradnog resursa.



Broj jedinica u odgovarajućem stupcu matrice I veći je od 1.

Zaglavljenje

Daljnji tijek operacija u sustavu (ili u dijelu sustava) je onemogućen.



Ostali problemi: ograničenost spremnika, kvar strojeva, ...

Zadatak ?!

Onemogućiti konfliktnu situaciju i zaglavljenje sustava.

↓

Upravljanje fleksibilnim proizvodnim sustavom.

Upravljanje:

- odrediti nedozvoljena stanja sustava,
- osigurati da sustav ne može ući u nedozvoljena stanja (stabilnost),
- odrediti kriterij koji sustav mora zadovoljiti (min vrijeme, max iskoristivost ...),
- pronaći algoritam upravljanja kojim se ostvaruje zadani kriterij.

Upravljačkim signalima promijeniti skup dopuštenih stanja tako da se u svakom stanje ostvare zadana svojstva => upravljane Petrijeve mreže.

Upravljivi događaji – moguće djelovanje upravljačkim algoritmom.

Neupravljivi događaji – ne može se djelovati upravljačkim algoritmom.

Pretpostavke:

predpražnjenje (*no preemption*) - resurs ne može prekidati posao koji radi i započeti novi, sve dok prvi posao nije dovršen,

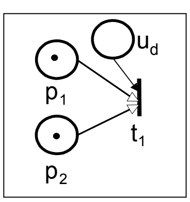
međusobno isključivanje (*mutual exclusion*) - višeradni resurs može istovremeno obavljati samo jedan zadatak (jedna operacija isključuje druge),

držanje (*hold while waiting*) - određena operacija “drži” resurse, koji su joj već pridjeljeni, sve dok nema sve resurse potrebne za njezino obavljanje,

nema kvarova (*no machine failure*) – strojevi su idealni i ne može doći do zaglavljenja ili prestanka rad uslijed kvara na stroju,

upravljivost i osmotrivost (*controllability and observability*) – svi događaji se mogu kontrolirati i sve komponente vektora stanja su mjerljive.

Upravljane Petrijeve mreže

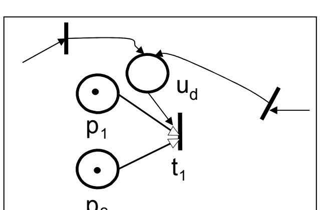


Kako spriječiti propaljivanje prijelaza t_1?

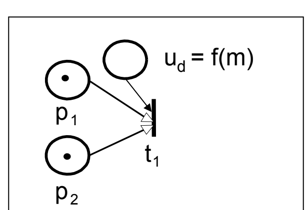
Dodati upravljačko mjesto u_d kao ulazno mjesto u prijelaz t_1.

Dvije mogućnosti:

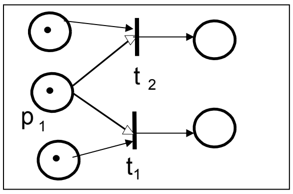
Upravljačko mjesto povezano s prijelazima Petrijeve mreže.



Upravljačko mjesto funkcija vektora stanja (ili nekog njegovog dijela).



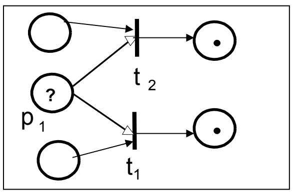
Onemogućavanje konflikta



paralelni konflikt

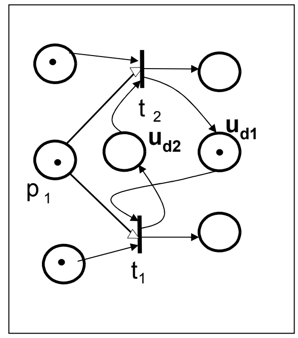
m(p₁)

$m_k = [\ \ldots\ 1\ \ldots\ ]^T$



m(p₁)

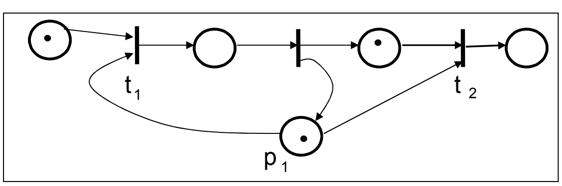
$m_{k+1} = [\ \ldots\ -1\ \ldots\ ]^T$

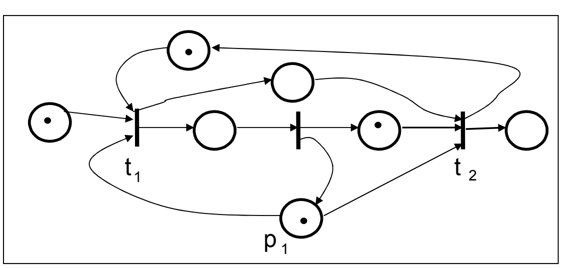


u_{d1} i u_{d2} čine p-invariantu

ulazi u upravljačka mjesta u_{d1} i u_{d2} ne moraju biti konfliktni prijelazi

senkvencijalni konflikt





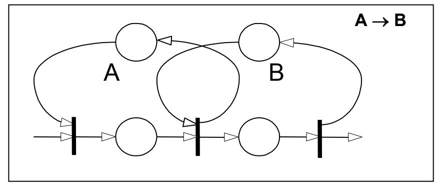
U određenim strukturama FPS-a sekvencijalni konflikt moguće je riješiti i samo jednim upravljačkim mjestom.

Onemogućavanje zaglavljenja

- onemogućavanje konflikta može (ali ne mora!) onemogućiti zaglavljenje.

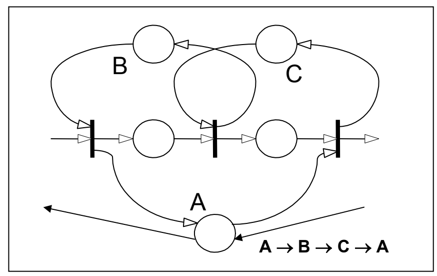
Kako dolazi do zaglavljenja?

Relacija čekanja *(wait relation):*
Kaže se da resurs r_j čeka r_k , $r_j \rightarrow r_k$, ako otpuštanju resursa r_j neposredno prethodi dostupnost resursa r_k .



Relacija kružnog čekanja *(circular wait relation):*
Za resurse u skupu CW={r_1, r_2, … ,r_n } za koje vrijedi $r_1 \rightarrow r_2 \rightarrow \ldots \rightarrow r_n \rightarrow r_1$, kaže se da su u kružnom čekanju.

Nužan, ali ne i dovoljan uvjet za postojanje kružnog čekanja jest postojanje barem jednog višeradnog resursa.



Kružno blokiranje *(circular blocking):*

Za neko kružno čekanje CW={r_i} kaže se da je u stanju kružnog blokiranja ako niti jedan resurs u CW ne može postati dostupan => *umrtvljeni resursi*.

nedostupnost resursa => neprisutnost oznaka => prazan sifon

sifon koji sadrži kružno čekanje, $CW \subseteq S$, naziva se *kritični sifon*

Kružno čekanje nije sifon !

Da bi sustav bio stabilan s obzirom na zaglavljenje, kritični sifon ne smije postati prazan.

Pitanje: Kako pronaći kritični sifon?

Odgovor: TEŠKO !

različiti algoritmi za različite strukture FPS-a => rekurzivne metode pretraživanja i eliminacije

Višeulazne proizvodne linije klase MRF_1

a) $p \in P$, $\bullet p \cap p\bullet = \emptyset$ – nema samostalnih petlji (čiste Petrijeve mreže),

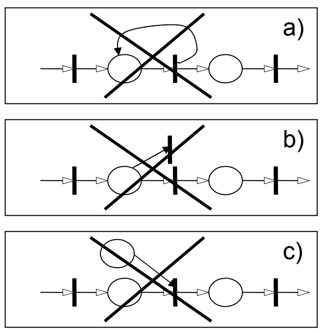
b) $p \in J$, $|p\bullet| = 1$ – nema poslova za koje je obavljanje sljedećeg posla slobodan izbor,

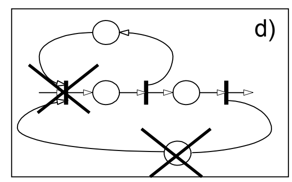
c) $\forall p_1, p_2 \in J$, $p_1 \neq p_2$, $p_1\bullet \cap p_2\bullet = \emptyset$ – nema poslova sklapanja,

d) $p \in J$: $\bullet\bullet p \cap R = p\bullet\bullet \cap R = R(p)$, $|R(p)| = 1$ – svaki posao zahtijeva jedan i samo jedan resurs,

e) $\exists r \in R$: $|J(r)| > 1$ – postoje višeradni resursi,

f) put svakog djela ima čvrsto definiran početak i kraj.





Višeulazne proizvodne linije klase MRF_1

a) $p \in P$, $\bullet p \cap p\bullet = \varnothing$ – nema samostalnih petlji (čiste Petrijeve mreže),

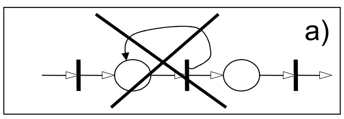
b) $p \in J$, $|p\bullet| = 1$ – nema poslova za koje je obavljanje sljedećeg posla slobodan izbor,

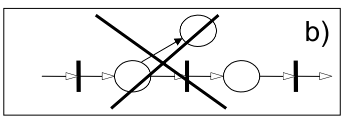
c) $\forall p_1, p_2 \in J$, $p_1 \neq p_2$, $p_1\bullet \cap p_2\bullet = \varnothing$ – nema poslova sklapanja,

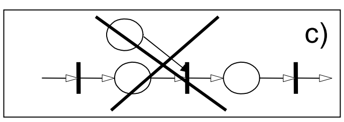
d) $p \in J : \bullet\bullet p \cap R = p\bullet\bullet \cap R = R(p)$, $|R(p)| = 1$ – svaki posao zahtijeva jedan i samo jedan resurs,

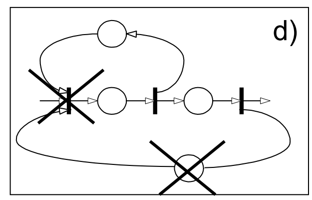
e) $\exists r \in R : |J(r)| > 1$ – postoje višeradni resursi,

f) put svakog djela ima čvrsto definiran početak i kraj.



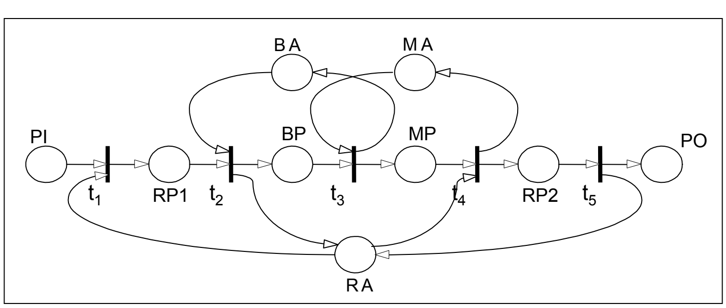






Određivanje kritičnog sifona u MRF_1

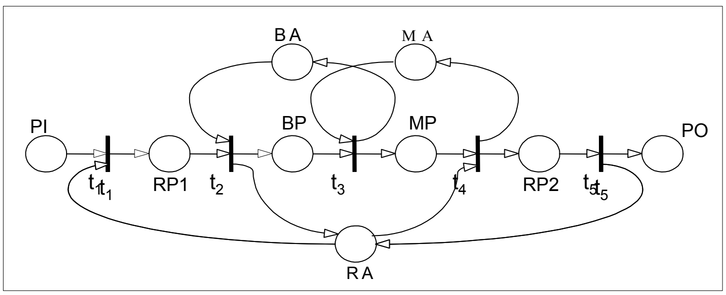
- kritični sifon sadrži kružno čekanje => pronaći kružna čekanja



Određivanje kritičnog sifona u MRF_1

- kritični sifon sadrži kružno čekanje => pronaći kružna čekanja
- prema definciji sifona $\bullet S \subset S\bullet$ => $\bullet C \not\subset C\bullet$ C nije sifon ! $C \cup ? = S$
- pronaći ulazne prijelaze u C koji nisu "blokirani" resursima u C

$$T_C^i = \bullet C$$
$$T_C = \bullet C \cap C \bullet$$
$$\longrightarrow \quad T_S = T_C^i \setminus T_C$$



$C = \{RA, BA, MA\}$

$\bullet C = \{ t_2, t_3, t_4, t_5 \}$

$C\bullet = \{ t_1, t_2, t_3, t_4 \}$

$T_C = \{ t_2, t_3, t_4 \}$

$T_S = \{ t_5 \}$

Određivanje kritičnog sifona u MRF_1

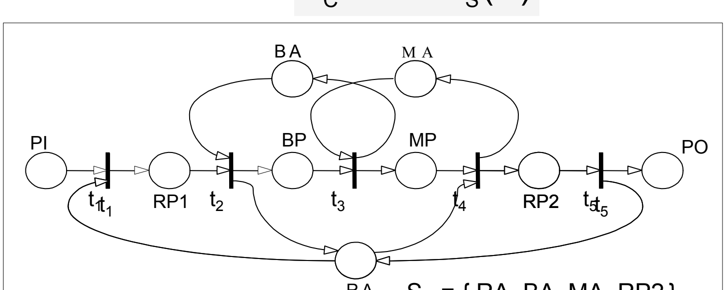
- pronaći ulazne prijelaze u C koji nisu "blokirani" resursima u C

$$T_S = T_C^i \setminus T_C$$

- pronaći mjesta koja "blokiraju" T_S i pripadaju skupu J(C)

$$\bullet T_S \cap J(C) = J_S(C)$$ Skup poslova sifona.

$$S_C = C \cup J_S(C)$$ Kritični sifon !



$S_C = \{ RA, BA, MA, RP2 \}$

$T_S = \{ t_5 \}$

$\bullet T_S = \{ RP2 \}$

$J(C) = \{ RP1, BP, MP, RP2 \}$

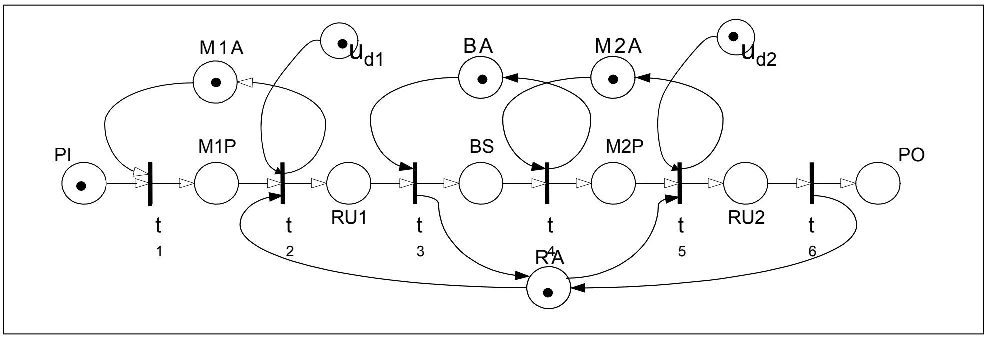
$\bullet T_S \cap J(C) = \{ RP2 \}$

$\bullet S_C = \{ t_2, t_3, t_4, t_5 \} \subset S_C\bullet = \{t_1, t_2, t_3, t_4, t_5 \}$

Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

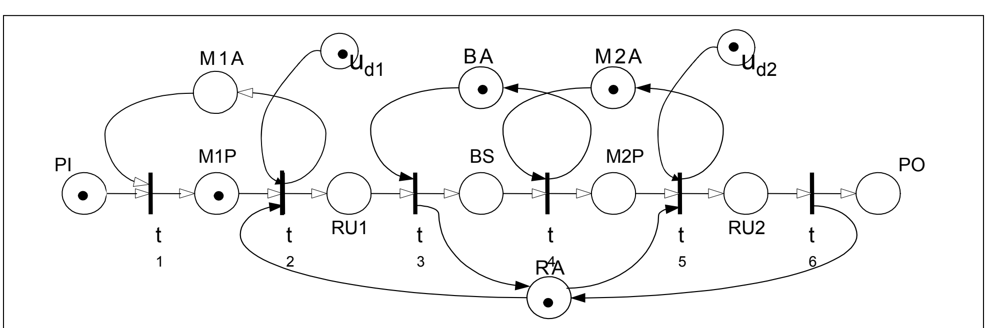
FBFS “first buffer first serve”,
prioritet punjenje FPS-a



Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

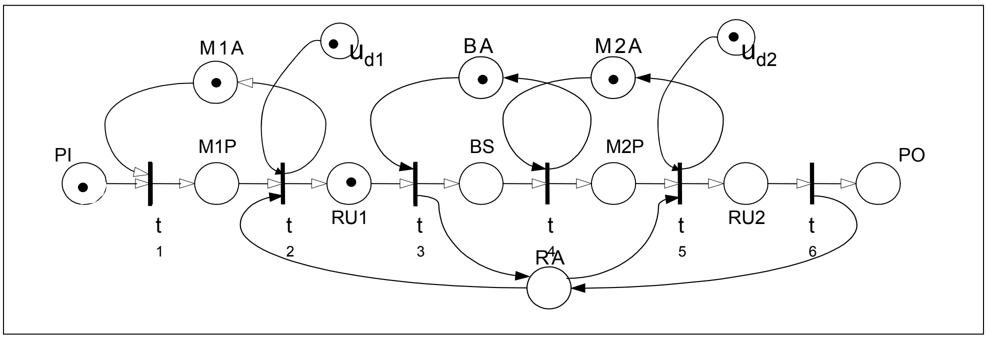
FBFS “first buffer first serve”,
prioritet punjenje FPS-a



Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

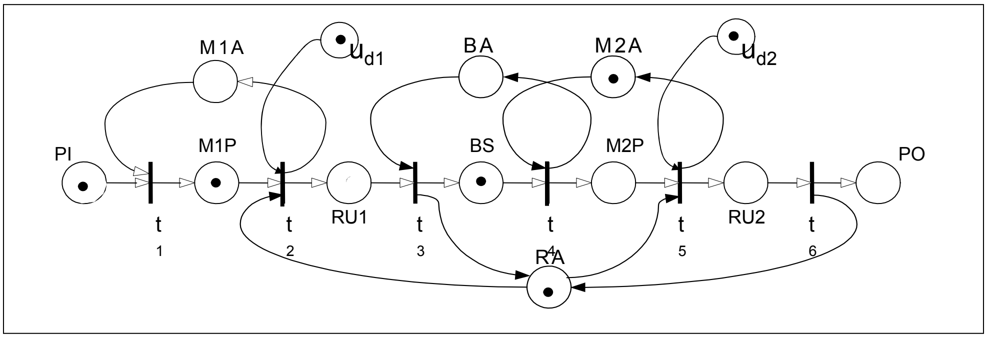
FBFS “first buffer first serve”,
prioritet punjenje FPS-a



Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

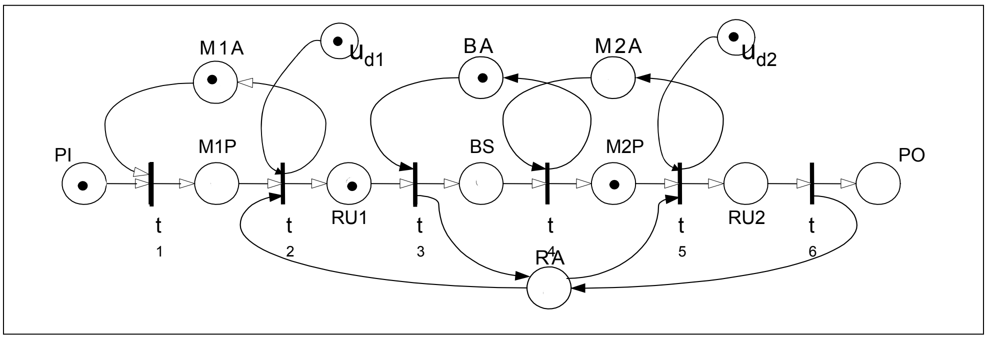
FBFS “first buffer first serve”,
prioritet punjenje FPS-a



Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

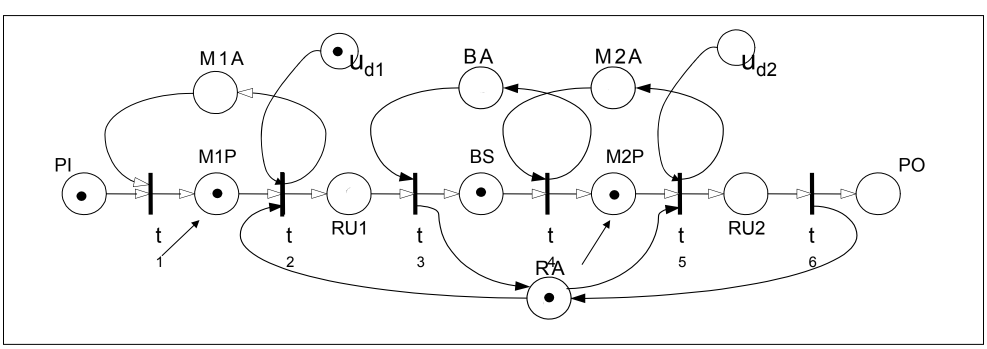
FBFS “first buffer first serve”,
prioritet punjenje FPS-a



Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

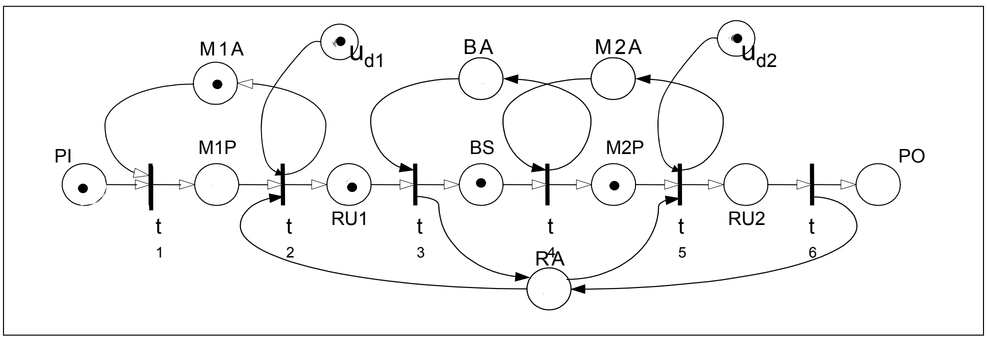
FBFS “first buffer first serve”,
prioritet punjenje FPS-a



Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

FBFS “first buffer first serve”,
prioritet punjenje FPS-a



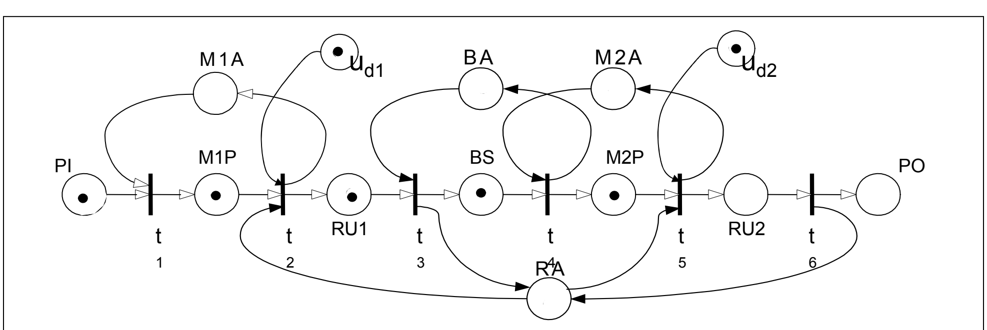
Primjer: zaglavljanje FPS-a

algoritam upravljanja: $m(u_{d1})=1$, $m(u_{d2})=0$ ako $m(M1P)+m(M2P)=2$
inače $m(u_{d1})=1$, $m(u_{d2})=1$

FBFS “first buffer first serve”,
prioritet punjenje FPS-a

kritični sifon je prazan => niti jedan prijelaz nije omogućen => FPS je zaglavljen !

Konflikt je onemogućen, no sustav se zaglavio.



Pitanje: kako odrediti upravljačke signale da kritični sifon ne postane prazan ?

Odgovor: uz malo matematike koja slijedi !

$T_C^o = C\bullet$

$T_C = \bullet C \cap C\bullet$

$\longrightarrow$ $T_Q = T_C^o \setminus T_C$

$T_Q\bullet \cap J(C) = J_Q(C)$ Skup poslova zamke

$Q_C = C \cup J_Q(C)$ Kritična zamka

$J_N(C) = J(C) \setminus \{J_Q(C) \cup J_S(C)\}$ Skup neutralnih poslova

$J_0(C) = \{J_Q(C) \setminus J_{SQ}(C)\} \cup J_N(C)$ Kritični podsustav ! $J_{SQ} = J_Q \cap J_S$

- može se pokazati da je:

$J(C) = J_S(C) \cup J_0(C)$ $\longrightarrow$ $m\{J(C)\} = m\{J_S(C)\} + m\{J_0(C)\}$

$m\{J(C)\} = m\{J_S(C)\} + m\{J_0(C)\}$

Podsjetnik!

$r_i \cup J(r_i) = p_{inv}$

Resurs sa skupom poslova koje obavlja, čini p-invariantu.

$\longrightarrow$ $C \cup J(C) = \bigcup_i p_{inv}^i$

$\downarrow$

$m\{C \cup J(C)\} = m_0(C) = \text{konst.}$

$\longleftarrow$ $m_0(C) = m(C) + m\{J(C)\}$

$\downarrow$

$m_0(C) = m(C) + m\{J_S(C)\} + m\{J_0(C)\}$

Podsjetnik!

$S_C = C \cup J_S(C)$

Kritični sifon !

$\downarrow$

$m_0(C) = m(S_C) + m\{J_0(C)\}$

Kritični sifon postat će prazan, $m(S_C) = 0$, ako i samo ako broj oznaka u kritičnom podsustavu, $m\{J_0(C)\}$, postane jednak početnom stanju broja oznaka kružnog čekanja, $m_0(C)$.

Primjer: zaglavljanje FPS-a (II)

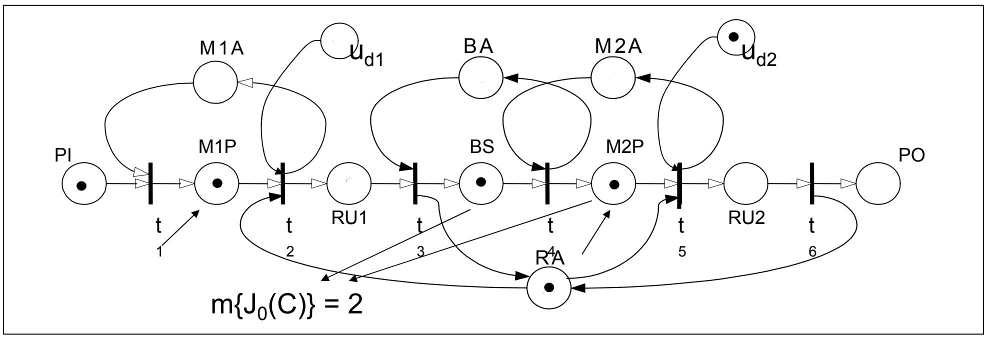
$J_0(C)$ = {RU1, BS, M2P}, $m_0(C) = 3$

algoritam upravljanja:

ako $m\{J_0(C)\} = 2$ onda $m(u_{d1})=0$, $m(u_{d2})=1$

inaće ako $m(M1P)+m(M2P)=2$ onda $m(u_{d1})=1$, $m(u_{d2})=0$

inače $m(u_{d1})=1$, $m(u_{d2})=1$



Primjer: zaglavljanje FPS-a (II)

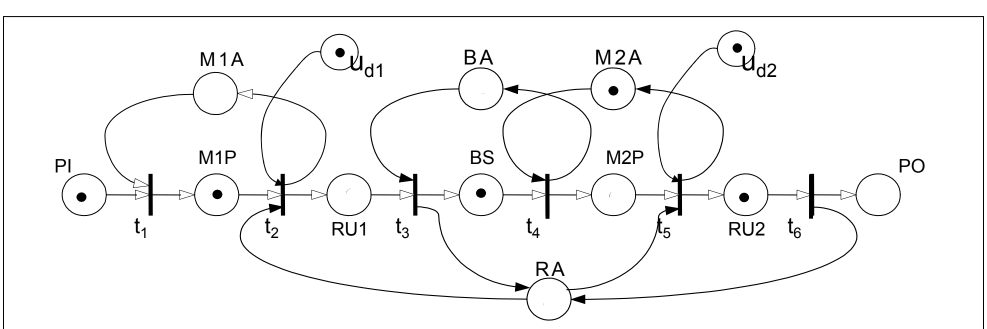
$J_0(C)$ = {RU1, BS, M2P}, $m_0(C) = 3$

algoritam upravljanja:

ako $m\{J_0(C)\} = 2$ onda $m(u_{d1})=0$, $m(u_{d2})=1$

inaće ako $m(M1P)+m(M2P)=2$ onda $m(u_{d1})=1$, $m(u_{d2})=0$

inače $m(u_{d1})=1$, $m(u_{d2})=1$



Pogled na stvar iz “drugog kuta”

- kružno čekanje C može se promatrati kao “distribucijski centar” zadataka,
- određivanje pojedinog zadatka mijenja broj slobodnih resursa,
- ideja je da su resursi što zaposleniji => nije problem ako nema slobodnog resursa
- problem je ako zauzeti resurs ne može postati slobodan,
- potrebno je “razmišljati” unaprijed => da li će određivanje pojedinog zadatka dovesti do kružnog blokiranja,

Sažetak: Postupak sinteze upravljačkog algoritma FPS-a

- odrediti operacije u FPS-u i pridijeliti im resurse,
- odrediti Petrijevu mrežu i graf Petrijeve mreže,
- odrediti strukturna svojstva Petrijeve mreže:

A) p-invarijante,
B) kružna čekanja,
C) konfliktne prijelaze,
D) kritične sifone,
E) kritične zamke,
F) kritične podsustave.

- odrediti maksimalni broj oznaka pojedinog kritičnog podsustava,
- definirati algoritam upravljanja koji će onemogućiti punjenje kritičnog podsustava,
- definirati algoritam upravljanja koji će onemogućiti konflikte,
- kombinacijom tih dvaju algoritama upravljati FPS.

p-invarijantno upravljanje (varijanta 2)

matrica događaja (*incidence matrix*)

P-invarianta je binarni P-vektor (vektor mjesta) za koji vrijedi: $W \cdot p = 0$

Vektori ***p-invarianti*** tvore matricu **P**: $P = [p_1 \quad p_2 \quad \ldots \quad p_n]$

$$W \cdot P = 0$$

Zadatak: ograničiti broj oznaka u nekom dijelu Petrijeve mreže

$$L \cdot m_p(k) \leq b$$

Skup nejednakosti (uvjeta) koji govori koliki je maksimalno dopušten broj oznaka (elementi vektora b) u skupinama mjesta (elementi vektora L*m_p)

$$L \cdot m_p(k) \leq b$$

U sustav dodajemo upravljačka mjesta – vektor $\mathbf{m_c}$, tako da vrijedi:

$$L \cdot m_p(k) + m_c(k) = b$$

Svakom uvjetu odgovara jedno upravljačko mjesto

Pitanje: kako upravljačka mjesta povezati s prijelazima Petrijeve mreže ?

$$W = [W_p \quad W_c]$$

Vektor stanja upravljane Petrijeve mreže (zatvoreni krug)

$$m = \begin{bmatrix} m_p \\ m_c \end{bmatrix}$$

Problem: odrediti $\mathbf{W_c}$ i $\mathbf{m_c(0)}$ koji zadovoljavaju zadane uvjete

Određivanje matrice regulatora $\mathbf{W_c}$

$$W \cdot P = 0 \longrightarrow \begin{bmatrix} W_p & W_c \end{bmatrix} \cdot P = 0$$

$$\begin{bmatrix} W_p & W_c \end{bmatrix} \cdot \begin{bmatrix} L^T \\ I \end{bmatrix} = 0$$

$$W_c = -W_p \cdot L^T$$

$$\begin{bmatrix} L^T \\ I \end{bmatrix} = P$$

$$L \cdot m_p(k) + m_c(k) = b \longrightarrow \begin{bmatrix} L & I \end{bmatrix} \cdot \begin{bmatrix} m_p(k) \\ m_c(k) \end{bmatrix} = b = \text{konst.}$$

Određivanje početnog stanja regulatora $\mathbf{m_c(0)}$

$$L \cdot m_p(k) + m_c(k) = b$$

za k=0 vrijedi:

$$L \cdot m_p(0) + m_c(0) = b$$

$$m_c(0) = b - L \cdot m_p(0)$$

Primjer:

m_p = [m(PI) m(M1P) m(M1A) m(M2P) m(M2A) m(PO)]^T

$m_p(0)$ = [5 0 3 0 2 0]^T

$$W_p = \begin{bmatrix} -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 \end{bmatrix}$$

Uvjet:

$$m(M1P) + m(M2P) \leq 3$$

b = [3]

$$L = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Jedan uvjet => $\mathbf{m_c}$ = [m_{c1}]

$m_c(0) = b - L \cdot m_p(0)$ => $\mathbf{m_c(0)}$ = [3]

$W_c = -W_p \cdot L^T$ => $W_c = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$