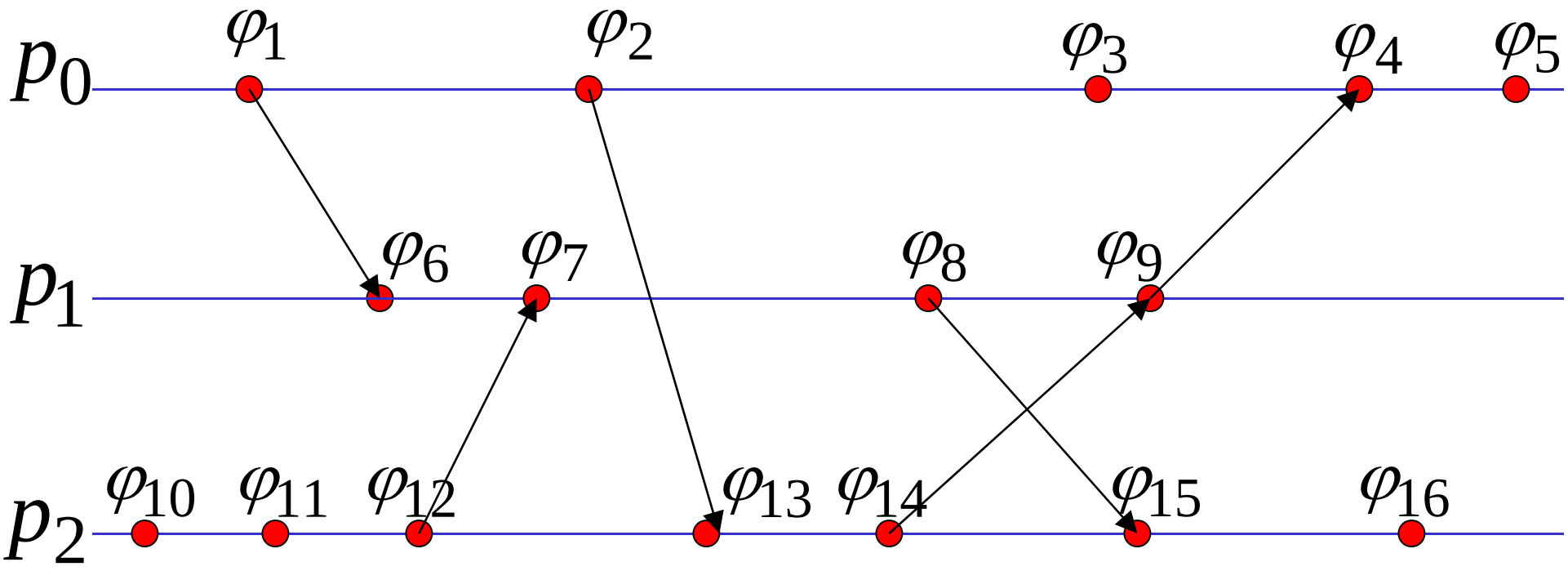
Uzročnost (Causality)

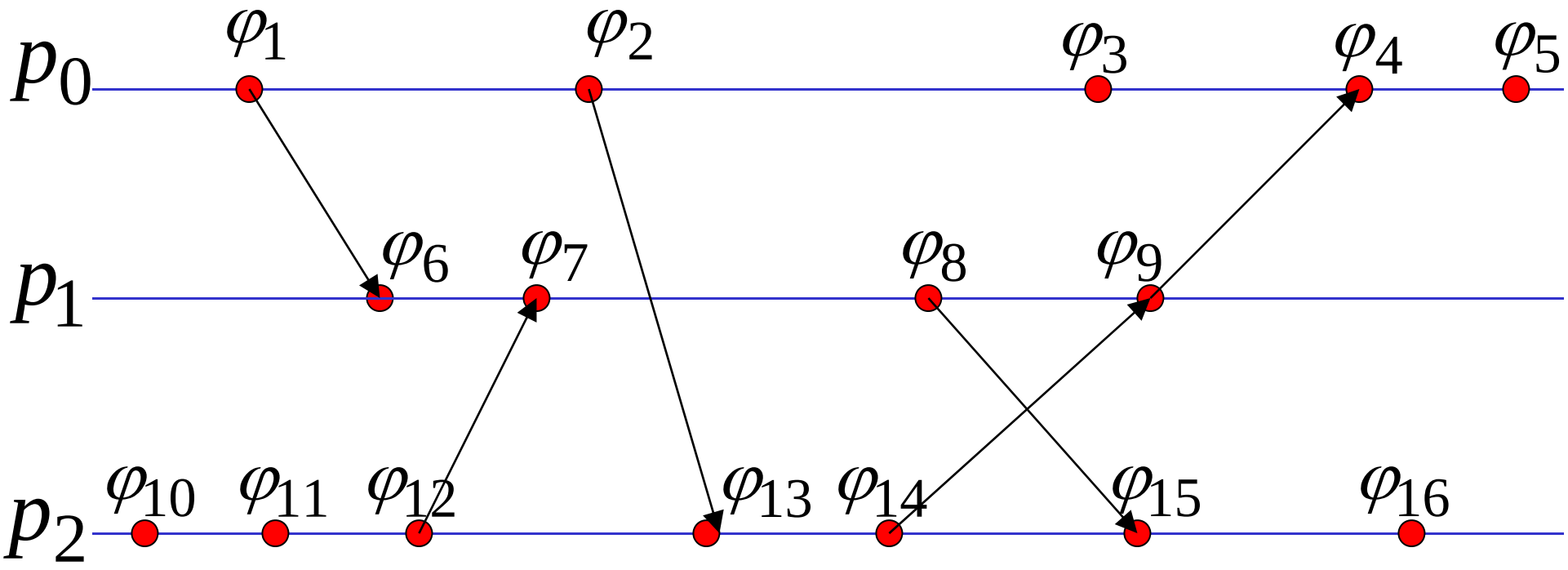
Relacija "desio se pre"



$$\varphi_1 \Rightarrow \varphi_2$$

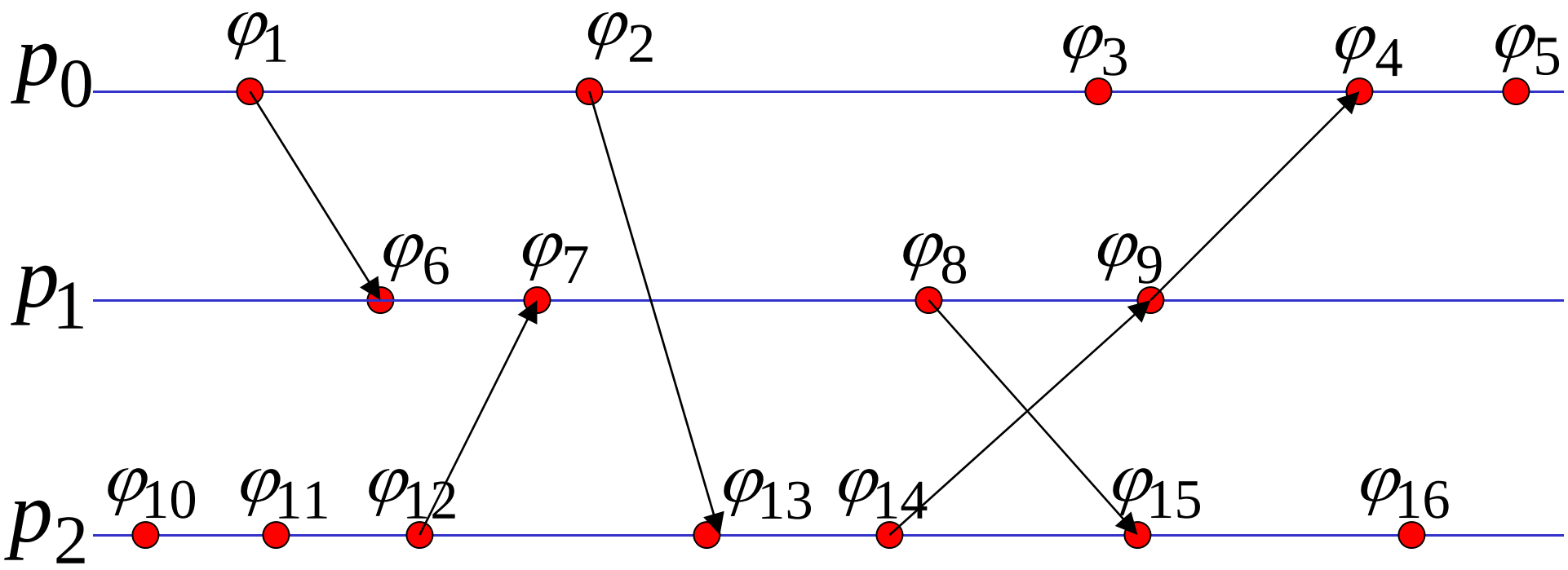
φ_1 desio se pre φ_2

(uzrokuje)



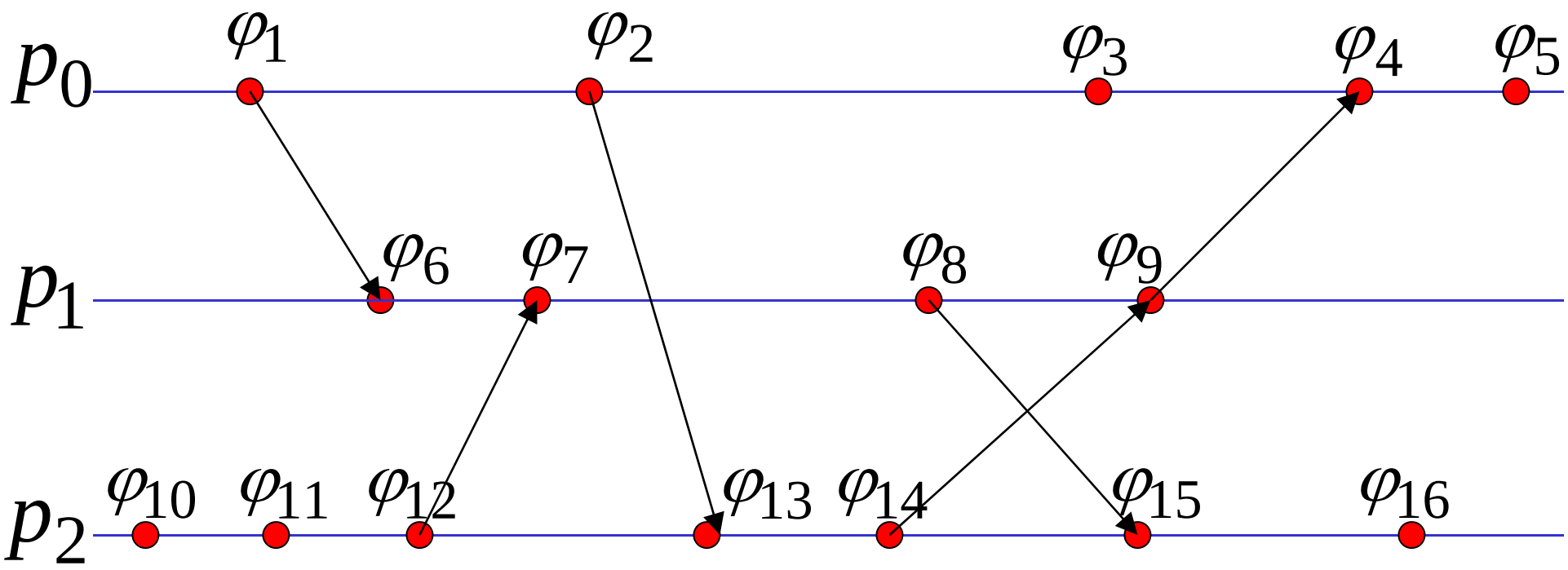
$$\varphi_1 \Rightarrow \varphi_6$$

Poruka je poslata od φ_1 ka φ_6



tranzitivnost

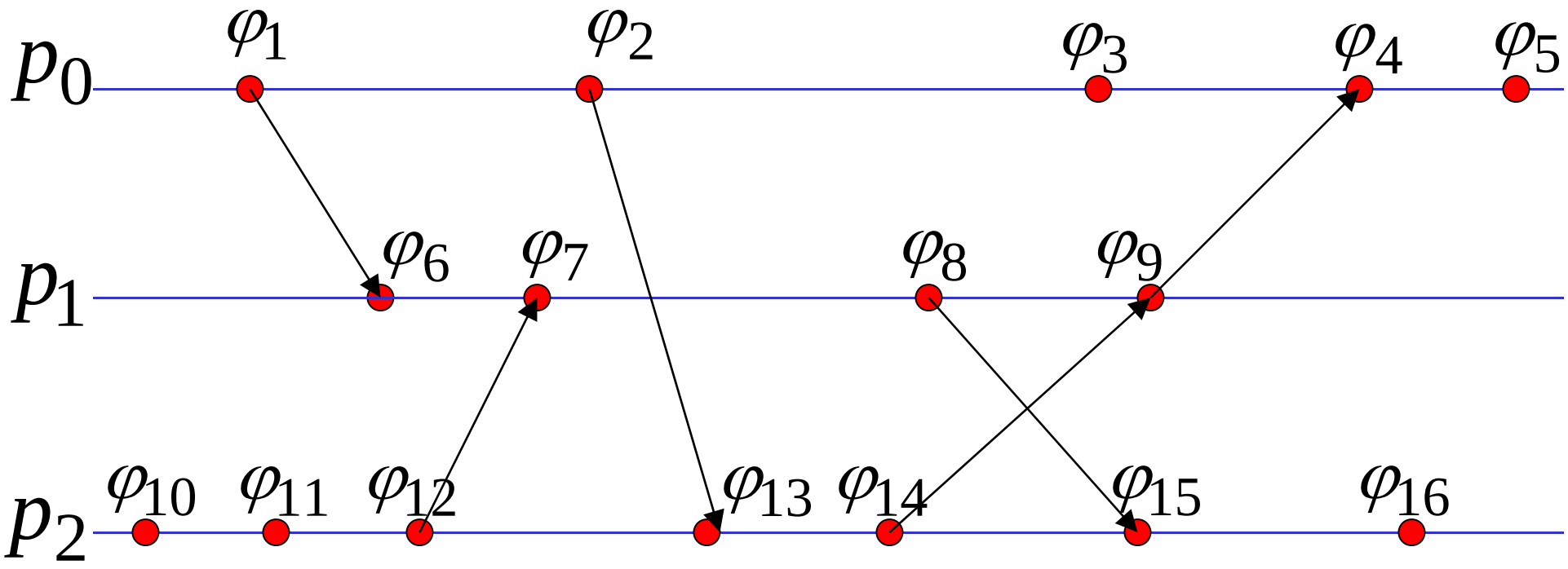
$$\varphi_1 \Rightarrow \varphi_6 \quad i \quad \varphi_6 \Rightarrow \varphi_7 \quad \Rightarrow \quad \varphi_1 \Rightarrow \varphi_7$$



$\varphi_1 \Rightarrow \varphi_8$ $\varphi_1 \Rightarrow \varphi_{16}$

$\varphi_{10} \Rightarrow \varphi_8$ $\varphi_2 \Rightarrow \varphi_9$

"desio se pre" je delimičan redosled



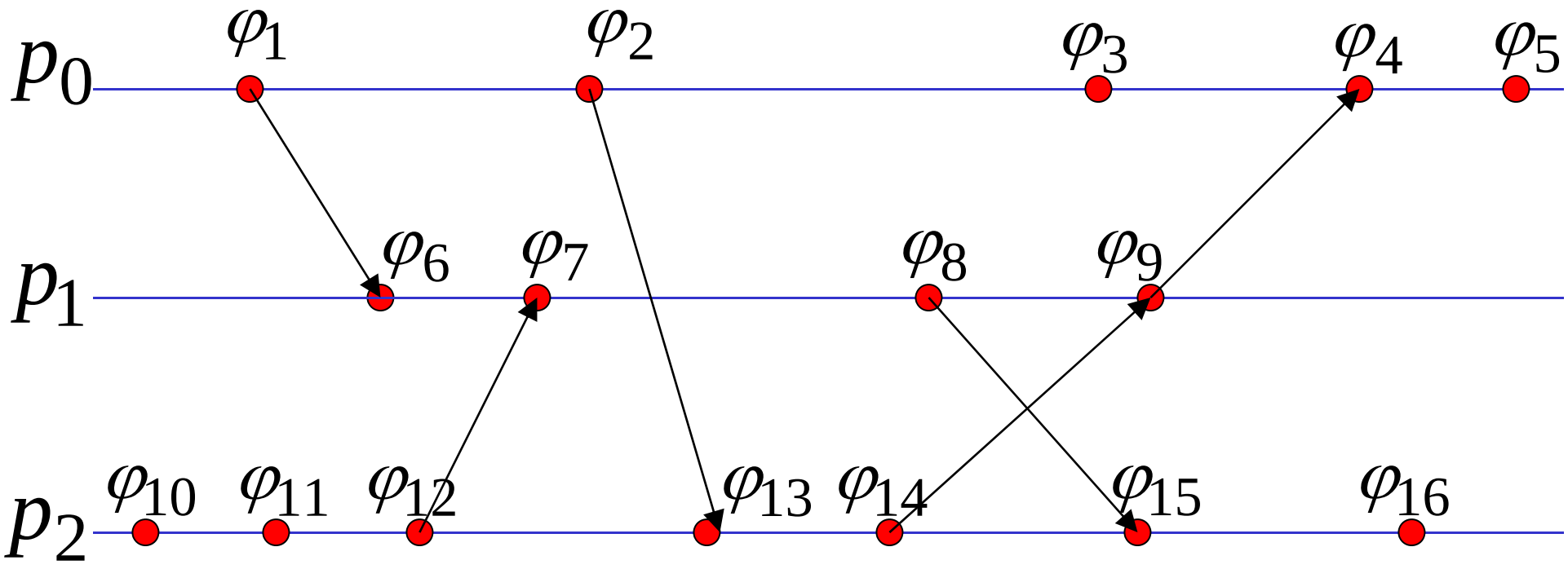
$\varphi_1 \Rightarrow \varphi_{10}$

Paralelni događaji: $\varphi_1 \quad \varphi_{10}$

$\varphi_{10} \Rightarrow \varphi_1$

$\varphi_1 \| \varphi_{10}$

"desio se pre" je delimičan redosled



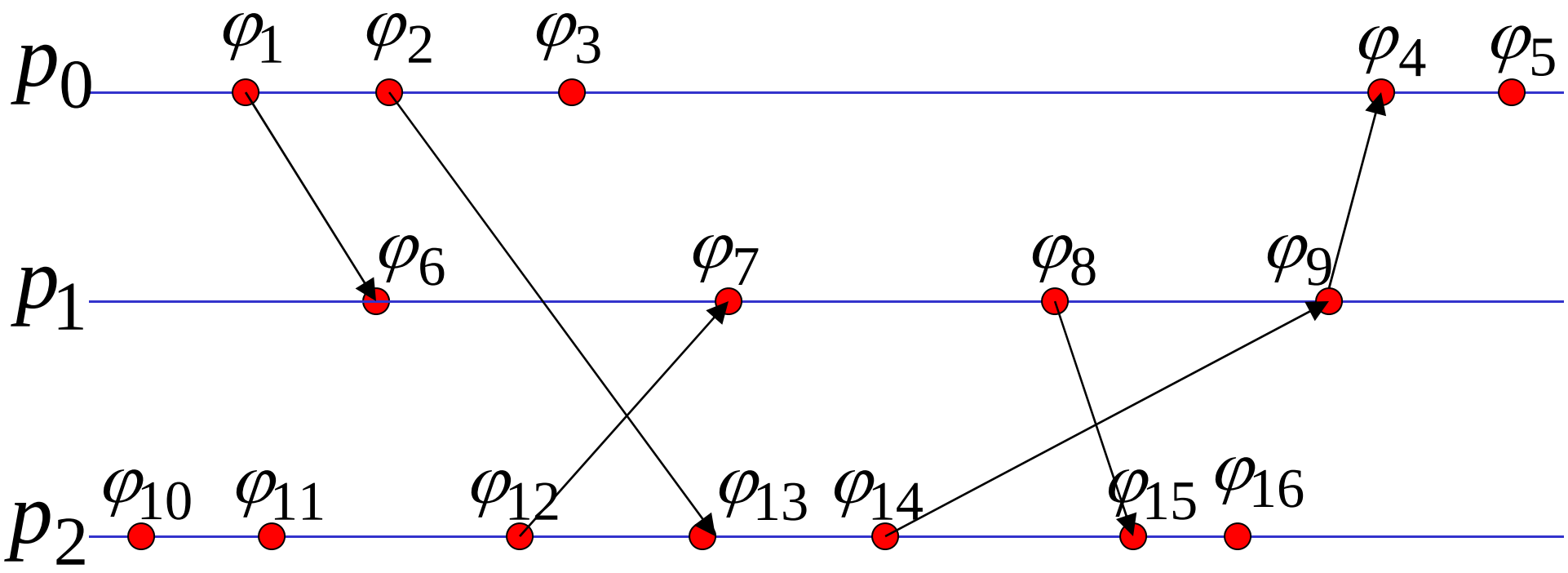
$\varphi_{13} \nRightarrow \varphi_8$

Paralelni događaji: $\varphi_8 \quad \varphi_{13}$

$\varphi_8 \nRightarrow \varphi_{13}$

$\varphi_8 \| \varphi_{13}$

Možemo pomerati paralelne događaje

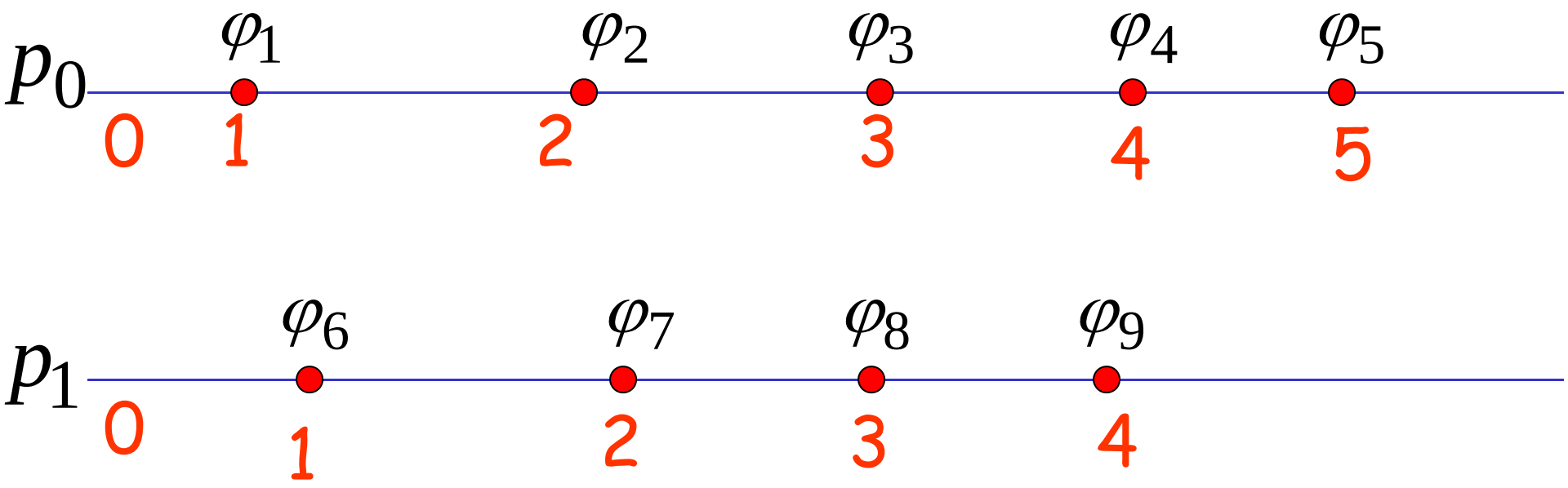


Relacija „desio se pre" se ne menja

Želimo da pronađemo mehanizam koji odslikava "desio se pre" relaciju

tako da se uzročnost može koristiti u raznim problemima računanja

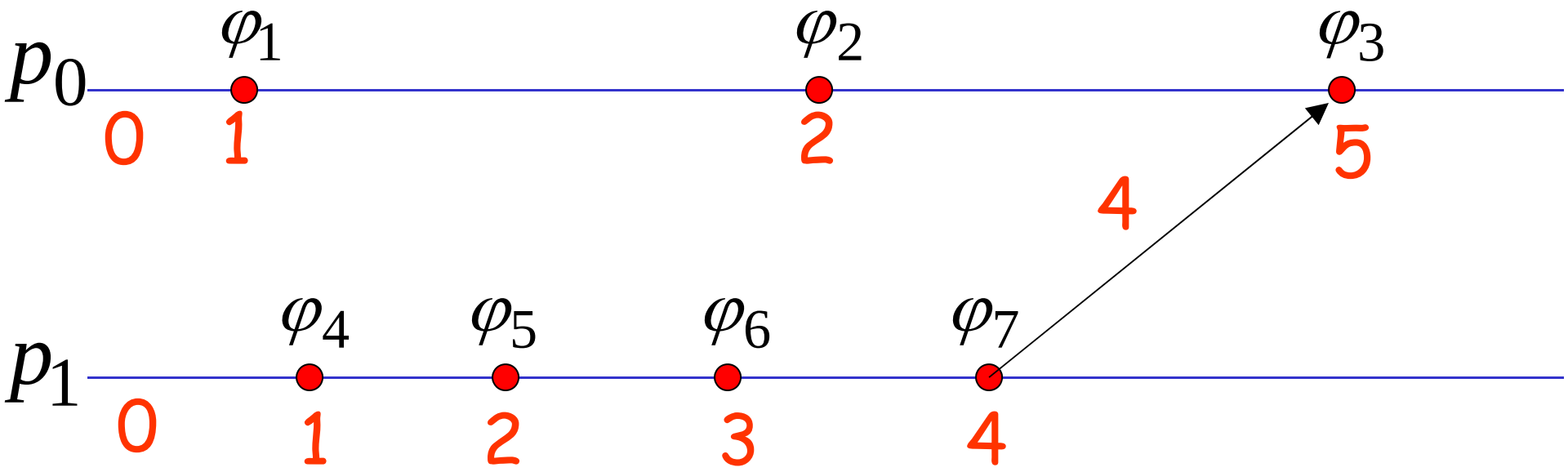
Logički satovi (Clocks)



U svakom procesu, logički sat se povećava za 1 za svaki lokalni događaj

$$LT(\varphi_1) = 1 \qquad LT(\varphi_2) = LT(\varphi_1) + 1 = 2$$

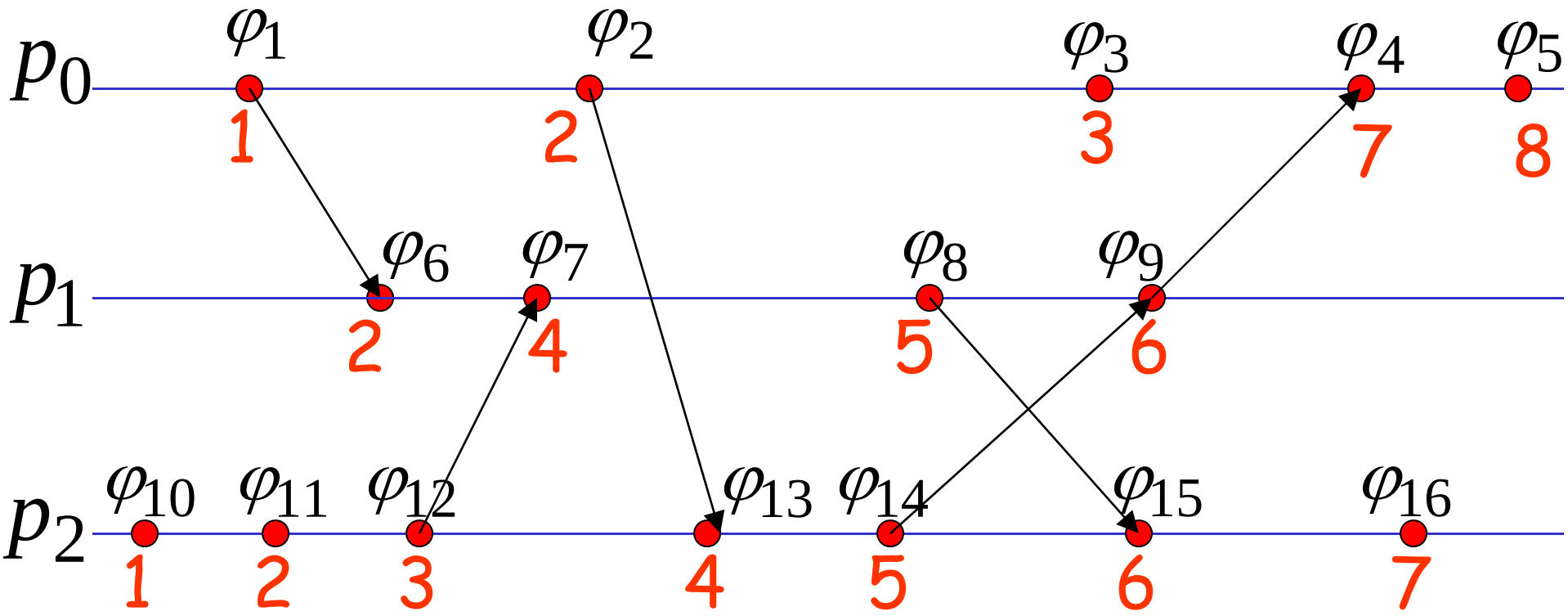
Logički satovi se prenose u porukama (piggy-packed)



$$LT(\varphi_3) = \max(LT(\varphi_2),\ LT(\varphi_7)) + 1$$

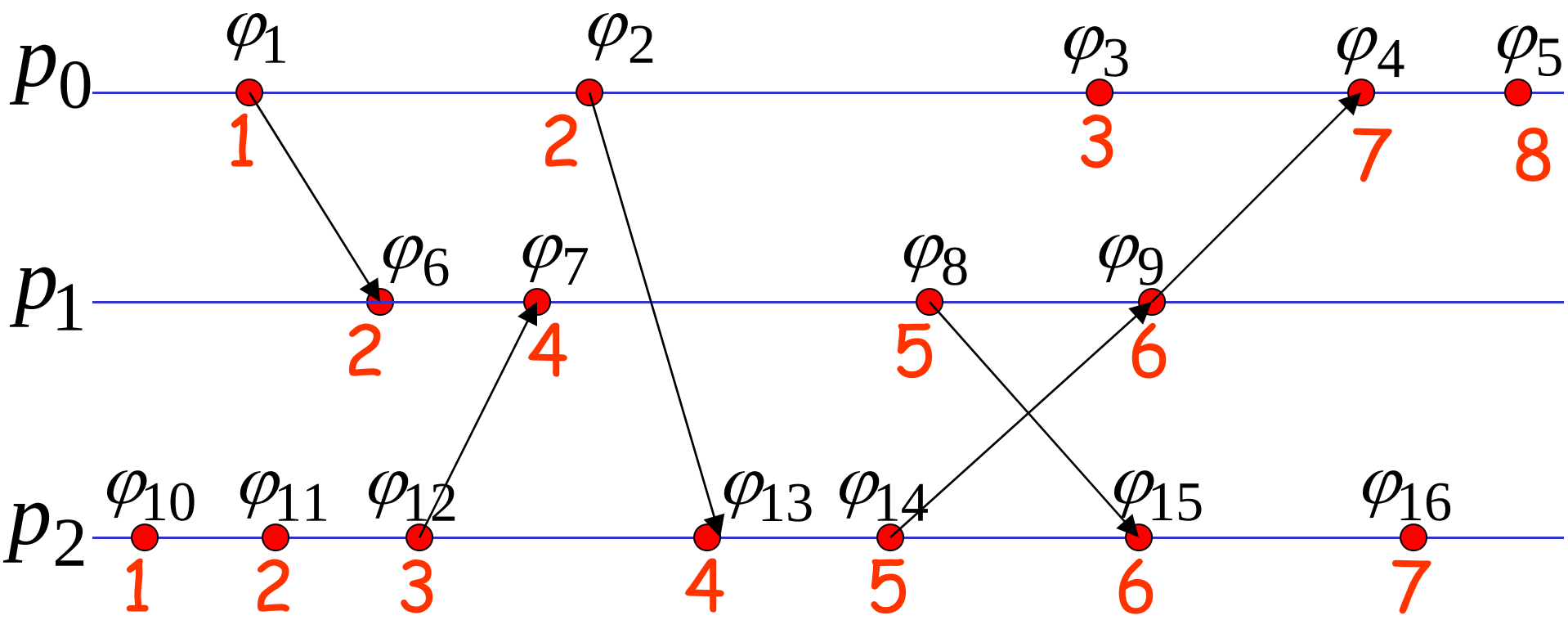
$$= \max(2,4) + 1$$

$$= 4 + 1 = 5$$



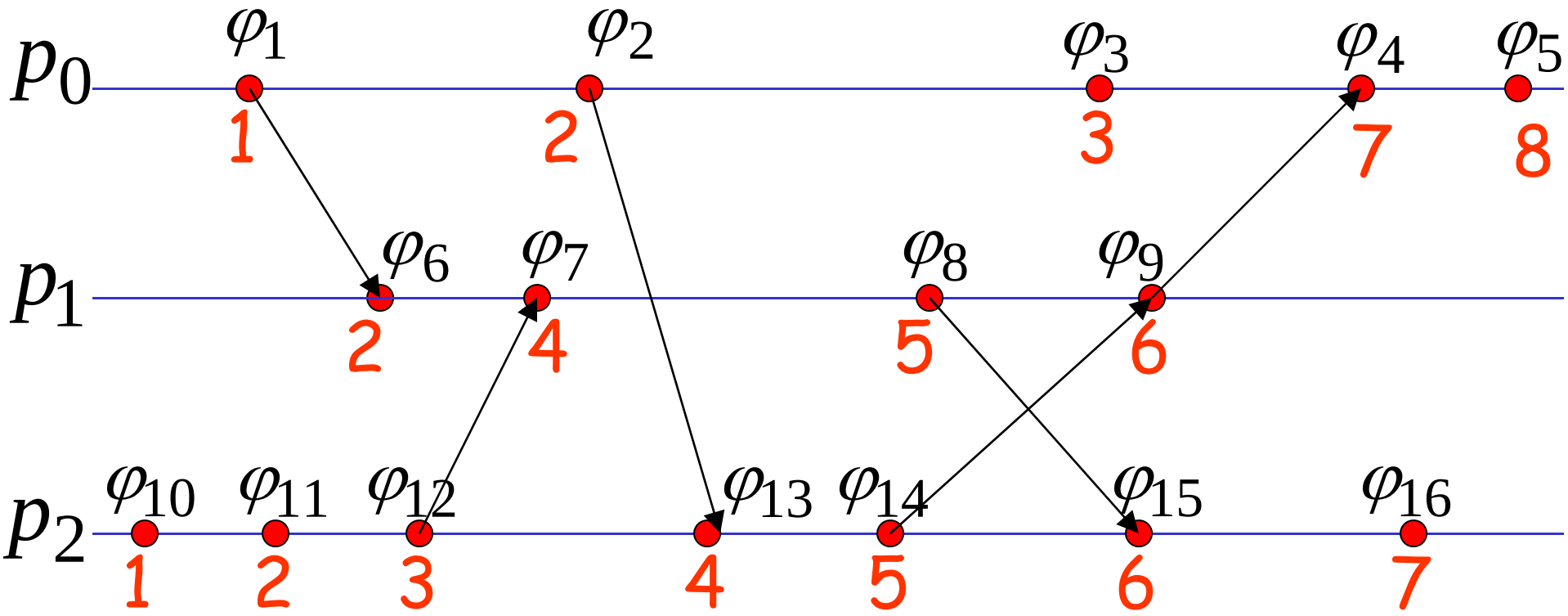
Izgleda kao da logički satovi odslikavaju desio se pre relaciju

$$\varphi_i \Rightarrow \varphi_j \quad \Rightarrow \quad LT(\varphi_i) < LT(\varphi_j)$$



Primer:

$$\varphi_1 \Rightarrow \varphi_7 \Longrightarrow LT(\varphi_1) = 1 < 4 = LT(\varphi_7)$$



Ipak, logički satovi nemogu odslikati paralelizam

$$LT(\varphi_{13}) = 4 < 5 = LT(\varphi_8)$$

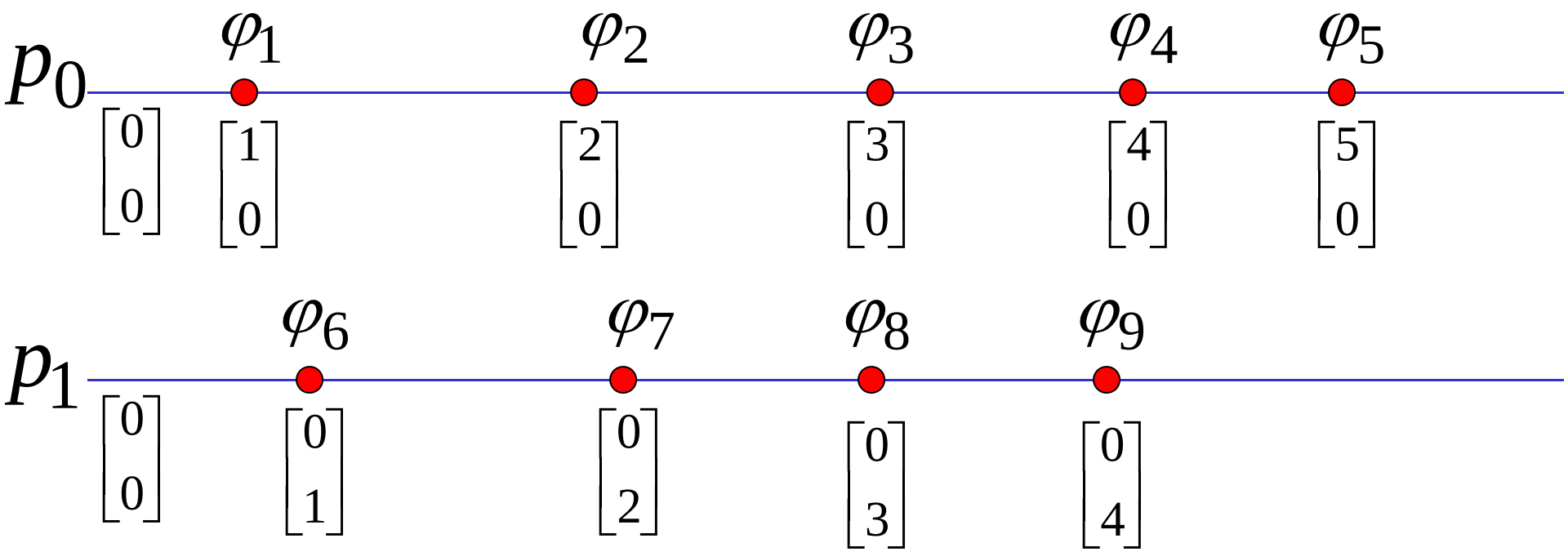
$$\varphi_8 \| \varphi_{13}$$

Paralelni događaji

$$(\varphi_{13} \Rightarrow \varphi_8 \,???)$$

Treba nam drugi mehanizam koji
može da odslika paralelizam događaja

Vektorski satovi (Vector Clocks)

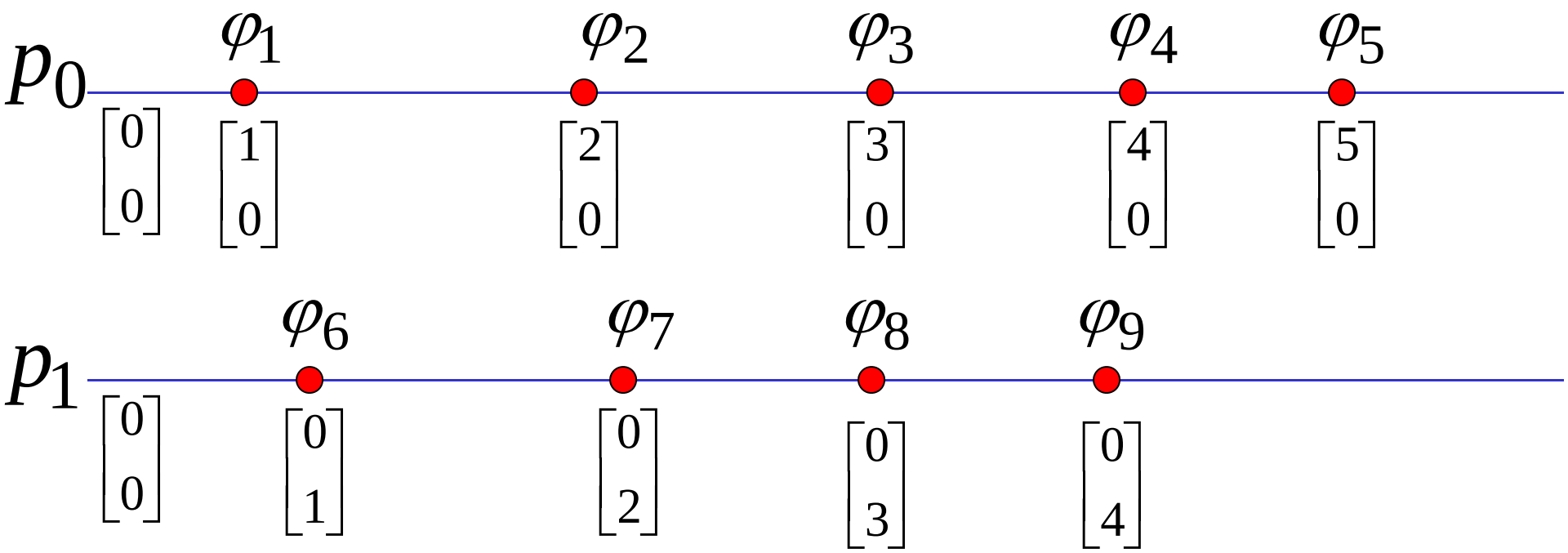


$$VC(\varphi_1) = \begin{bmatrix}1\\0\end{bmatrix}$$

Elem procesa p_0

Elem procesa p_1

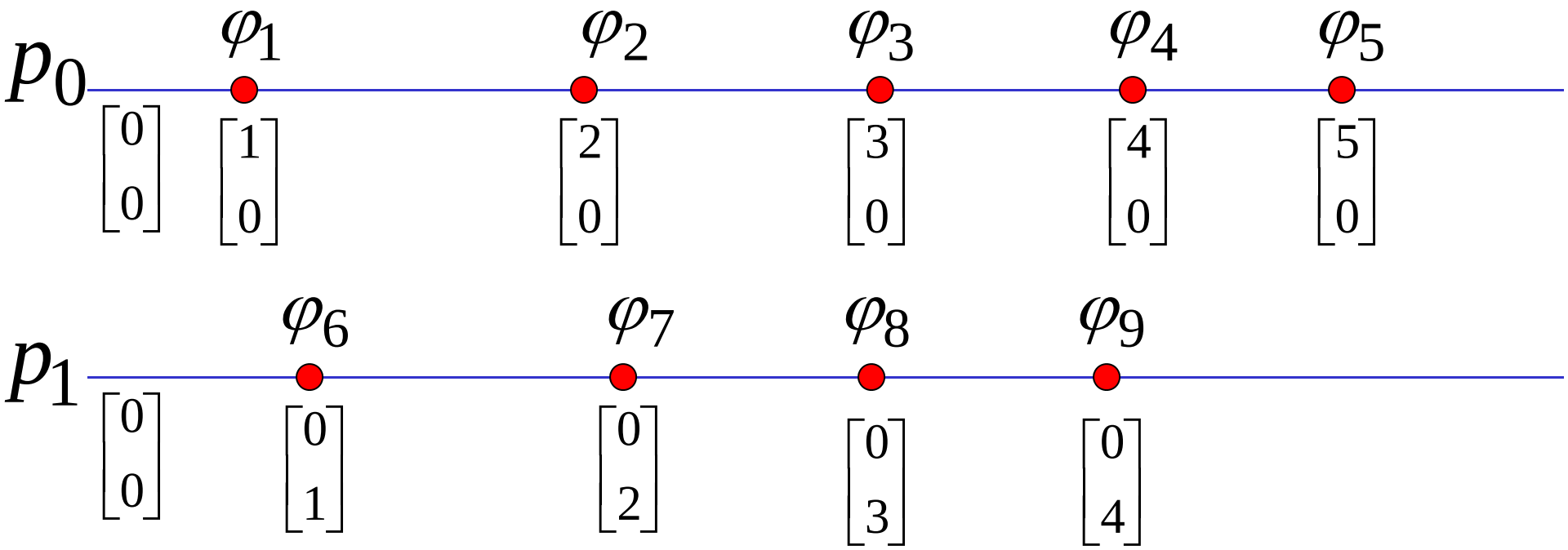
Svaki proces povećava svoj el. za svaki događaj



$$VC(\varphi_3) = VC(\varphi_2) + \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}2\\0\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}3\\0\end{bmatrix}$$

povećaj za 1

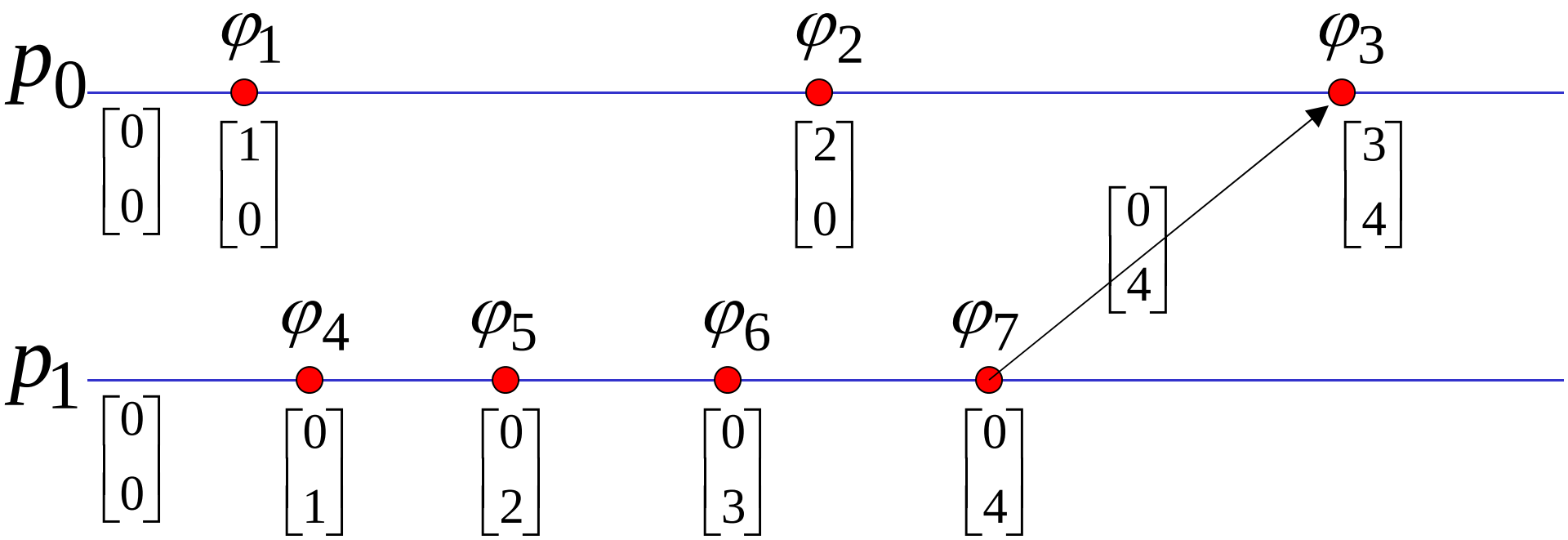
Svaki proces povećava svoj el. za svaki događaj



$$VC(\varphi_9) = VC(\varphi_8) + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\3\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\4\end{bmatrix}$$

povećaj za 1

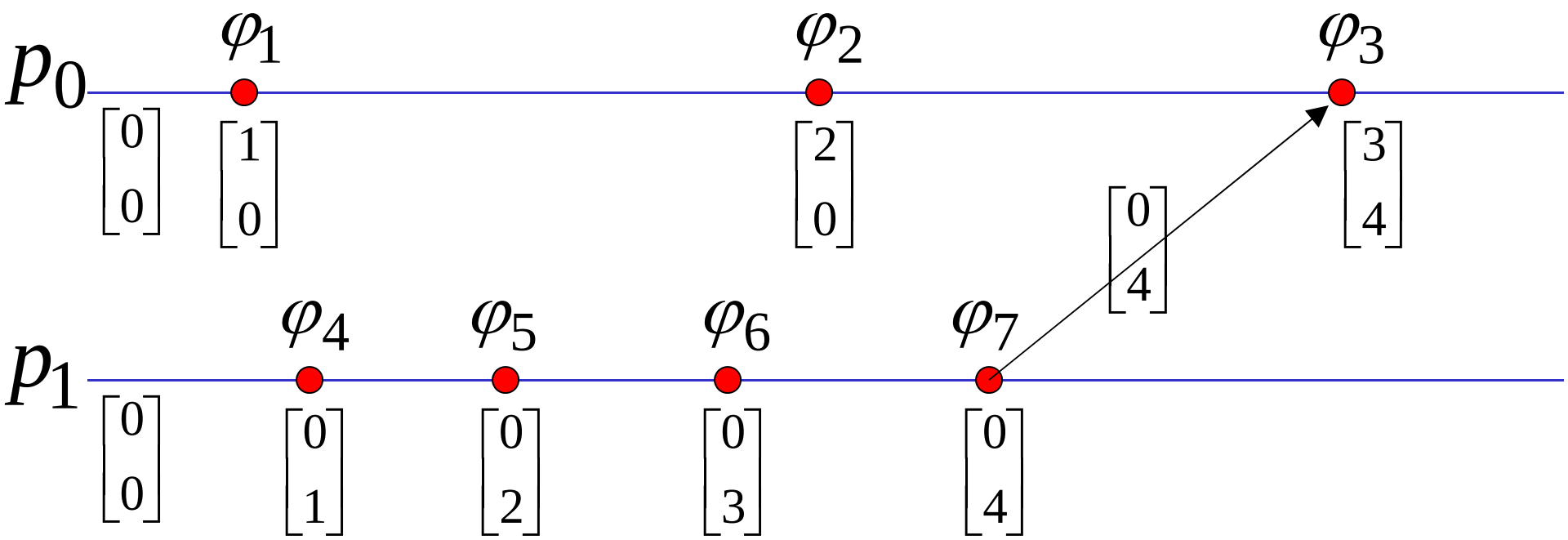
vektorski satovi se prenose u porukama



$$\max\left(\begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}\right) = \begin{bmatrix}2\\4\end{bmatrix}$$

Maksimum za svaki elem

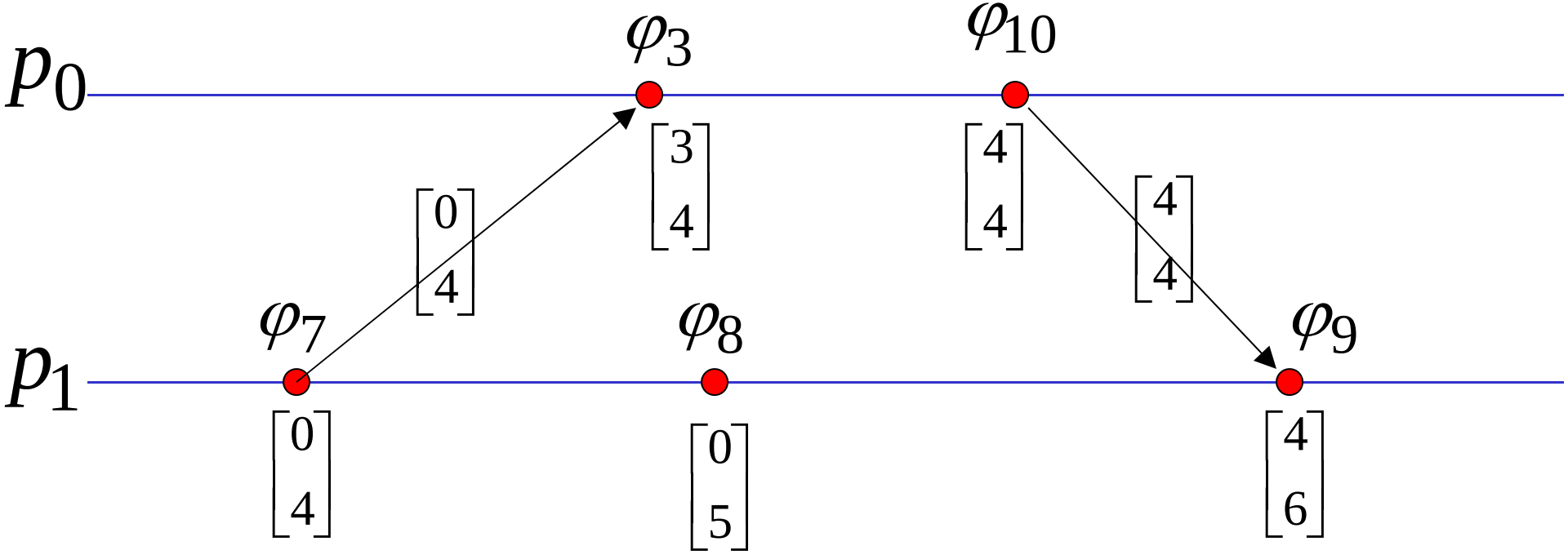
vektorski satovi se prenose u porukama



$$VC(\varphi_3) = \max(VC(\varphi_2), VC(\varphi_7)) + \begin{bmatrix}1\\0\end{bmatrix}$$

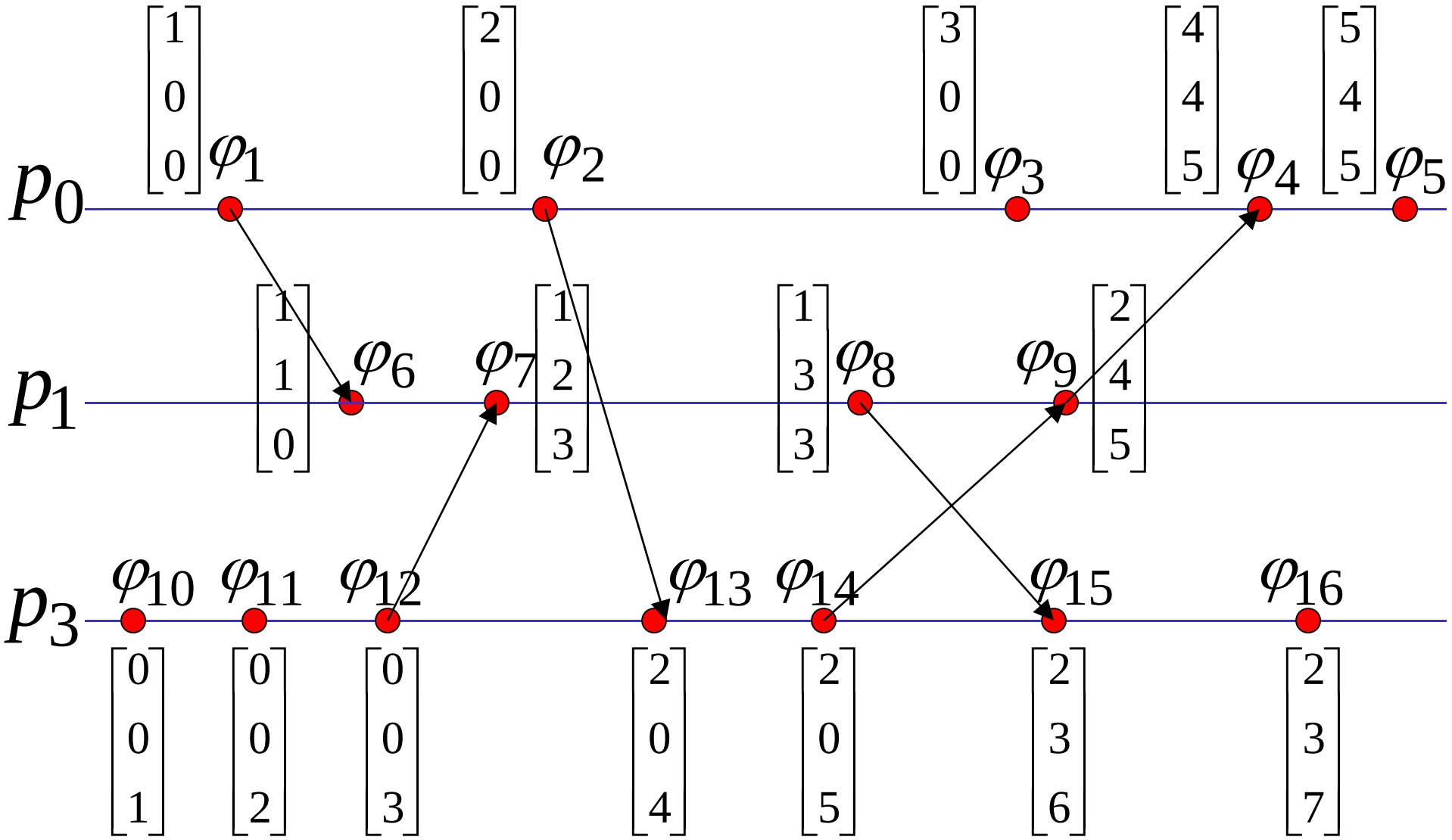
$$= \max\left(\begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}\right) + \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}2\\4\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}3\\4\end{bmatrix}$$

vektorski satovi se prenose u porukama



$$VC(\varphi_9) = \max(VC(\varphi_8), VC(\varphi_{10})) + \begin{bmatrix}0\\1\end{bmatrix}$$

$$= \max\left(\begin{bmatrix}0\\5\end{bmatrix}, \begin{bmatrix}4\\4\end{bmatrix}\right) + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}4\\5\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}4\\6\end{bmatrix}$$


p_0
p_1
p_3
φ_1 φ_2 φ_3 φ_4 φ_5
φ_6 φ_7 φ_8 φ_9
φ_{10} φ_{11} φ_{12} φ_{13} φ_{14} φ_{15} φ_{16}
[1 0 0] [2 0 0] [3 0 0] [4 4 5] [5 4 5]
[1 1 0] [1 2 3] [1 3 3] [2 4 5]
[0 0 1] [0 0 2] [0 0 3] [2 0 4] [2 0 5] [2 3 6] [2 3 7]

Poređenje vektorskih satova

Pišemo

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Ako $a_i \leq b_i$ za sve i

Primeri:

$$\begin{bmatrix} 3 \\ 7 \\ 4 \end{bmatrix} < \begin{bmatrix} 9 \\ 7 \\ 5 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} < \begin{bmatrix} 6 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 5 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 3 \end{bmatrix}$$

Neuporedivi vektorski satovi

Pišemo $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \not\leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

Ako nije $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$ niti $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \geq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

Primeri:

$$\begin{bmatrix} 3 \\ 7 \\ 4 \end{bmatrix} \not< \begin{bmatrix} 2 \\ 8 \\ 5 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \not< \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Vektorski satovi odslikavaju uzročnost

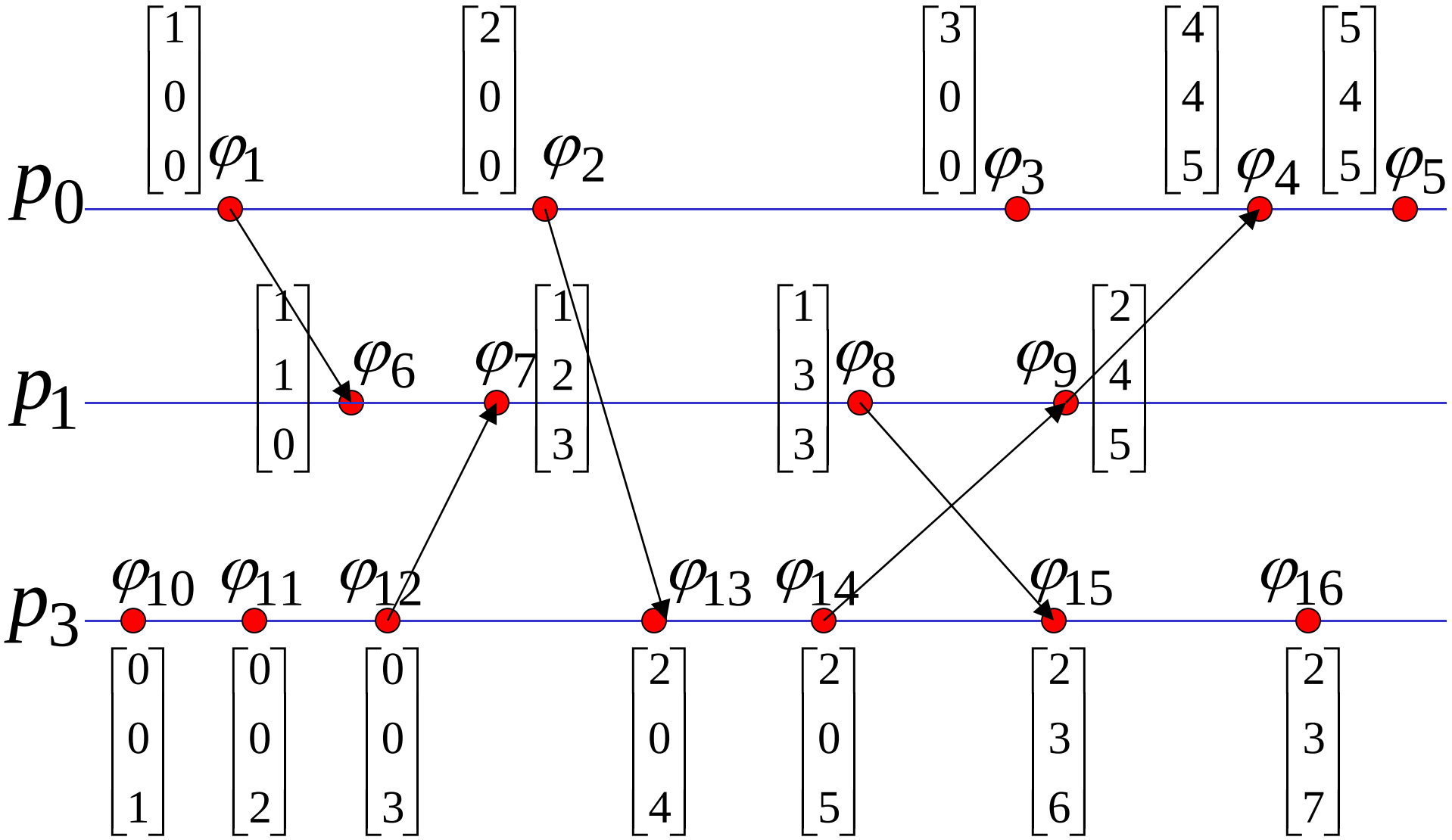
Ako $\varphi_1 \Rightarrow \varphi_2$ onda $VC(\varphi_1) < VC(\varphi_2)$

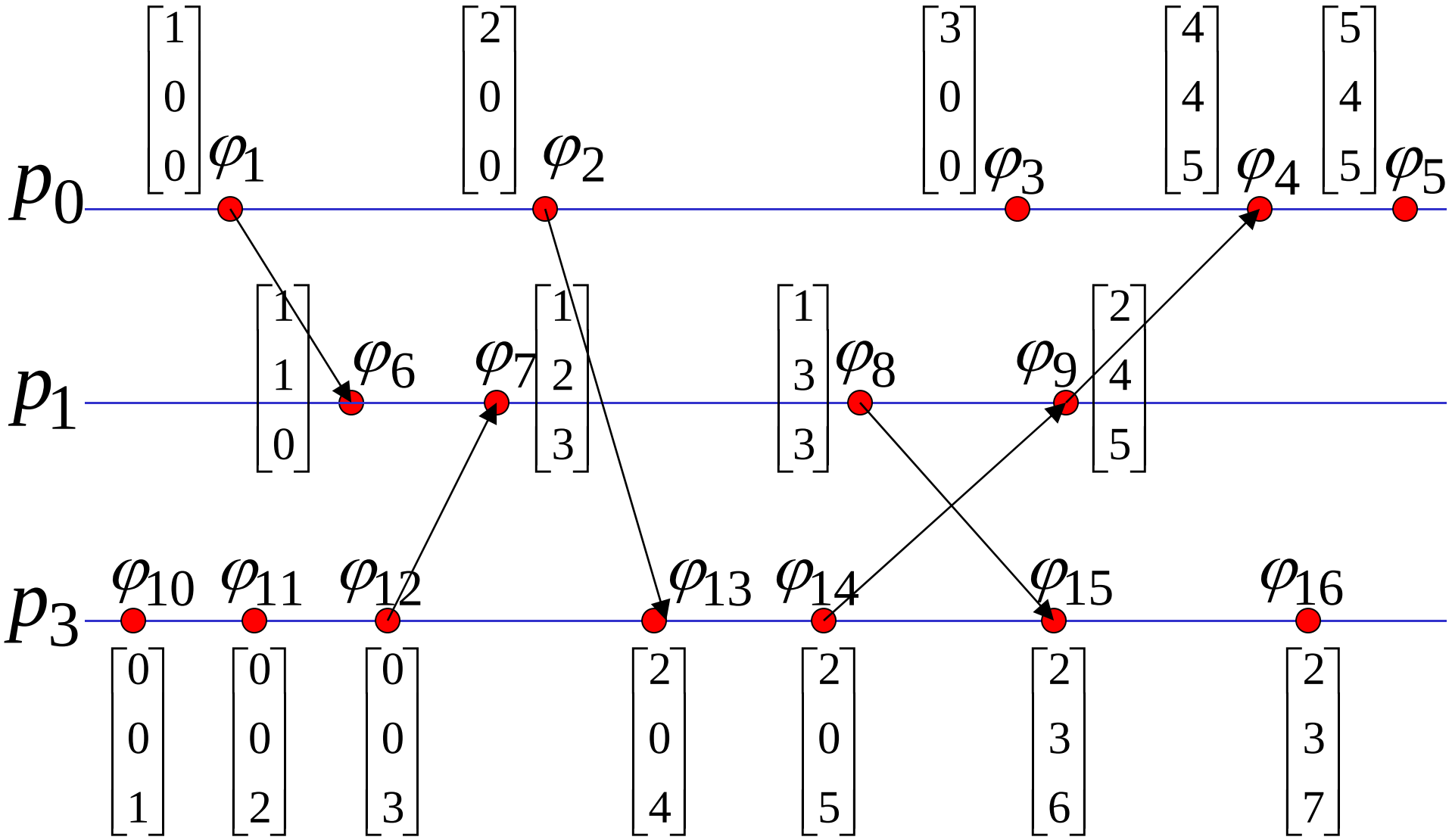
Ako $\varphi_1 \| \varphi_2$ onda $VC(\varphi_1) \not< VC(\varphi_2)$

Ispitivanjem vektorskih satova možemo odrediti redosled događaja

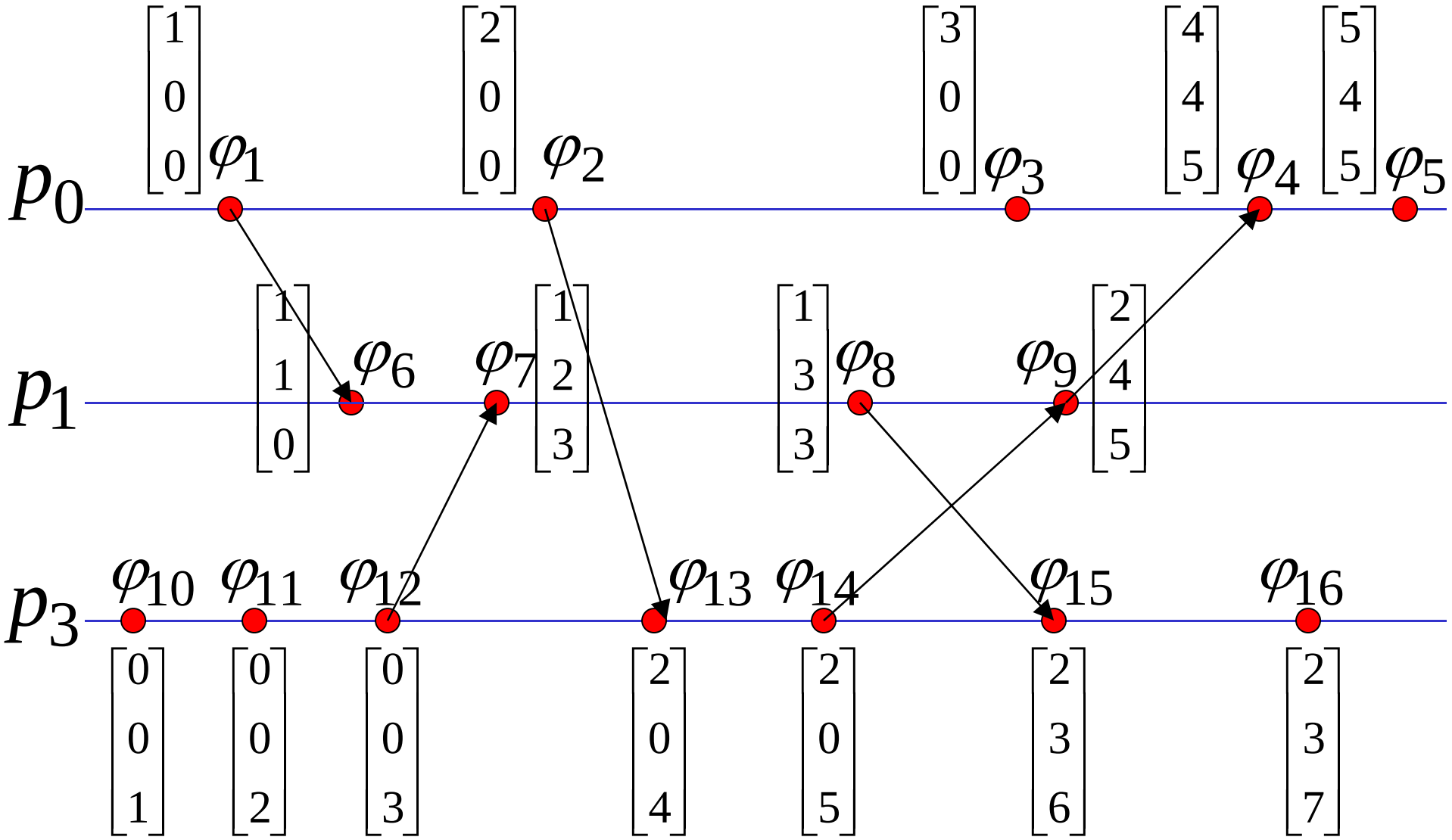
Ako $VC(\varphi_1) < VC(\varphi_2)$ onda $\varphi_1 \Rightarrow \varphi_2$

Ako $VC(\varphi_1) \not< VC(\varphi_2)$ onda $\varphi_1 \| \varphi_2$

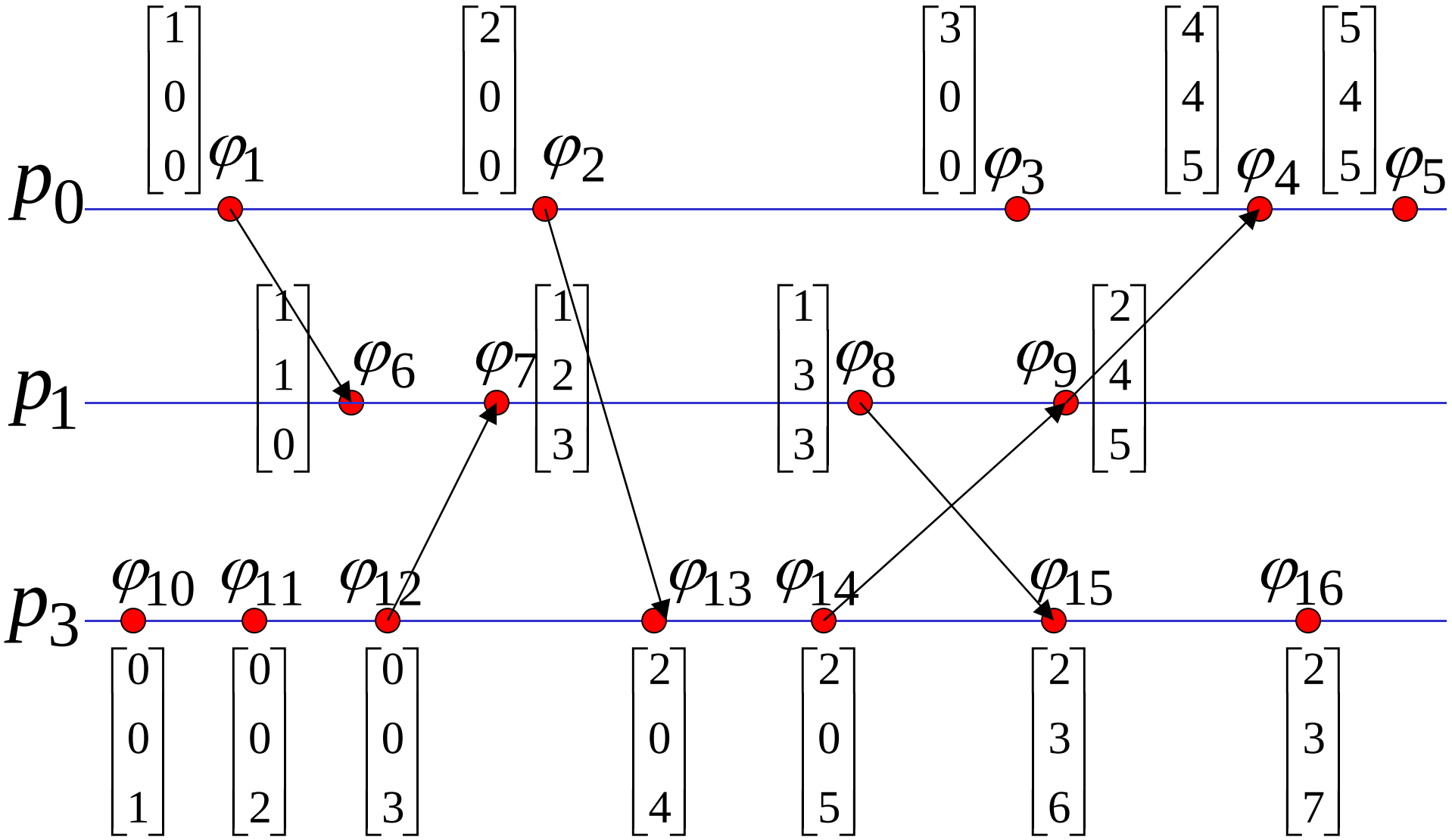

p_0
p_1
p_3
φ_1 φ_2 φ_3 φ_4 φ_5
φ_6 φ_7 φ_8 φ_9
φ_{10} φ_{11} φ_{12} φ_{13} φ_{14} φ_{15} φ_{16}
$\varphi_1 \Rightarrow \varphi_2$
$VC(\varphi_1) < VC(\varphi_2)$



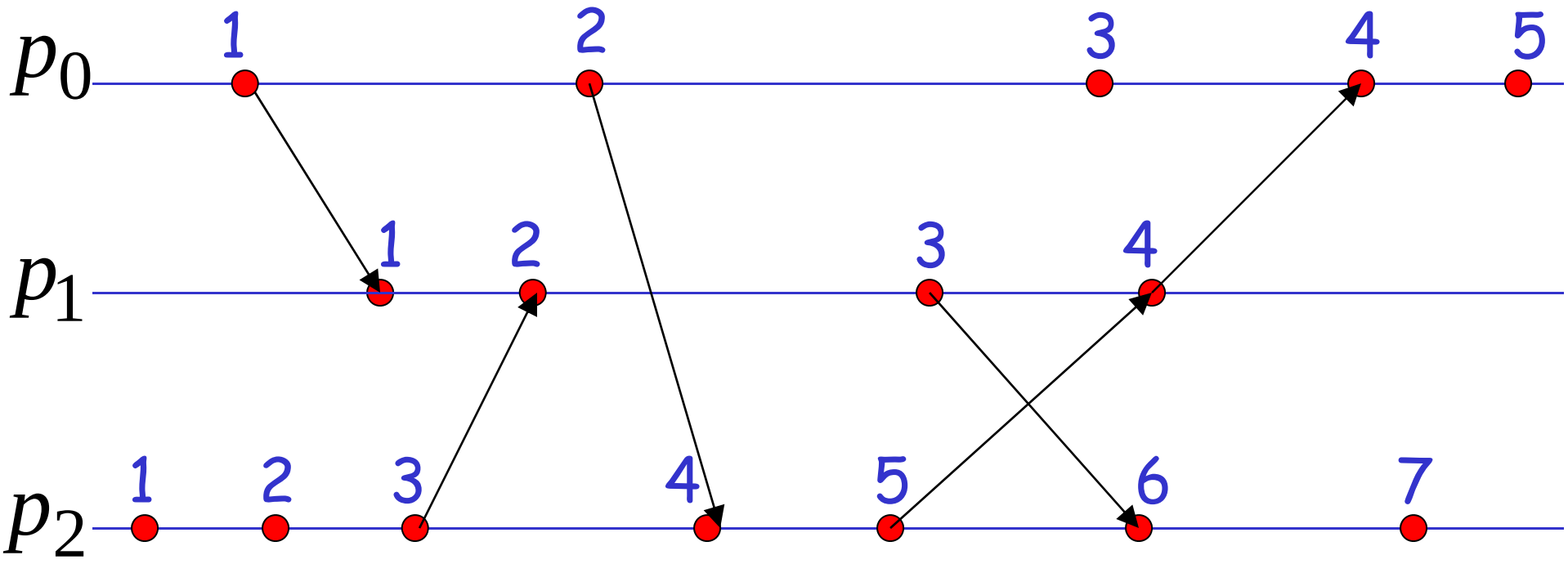
$$VC(\varphi_2) < VC(\varphi_{13}) \quad \Rightarrow \quad \varphi_2 \Rightarrow \varphi_{13}$$



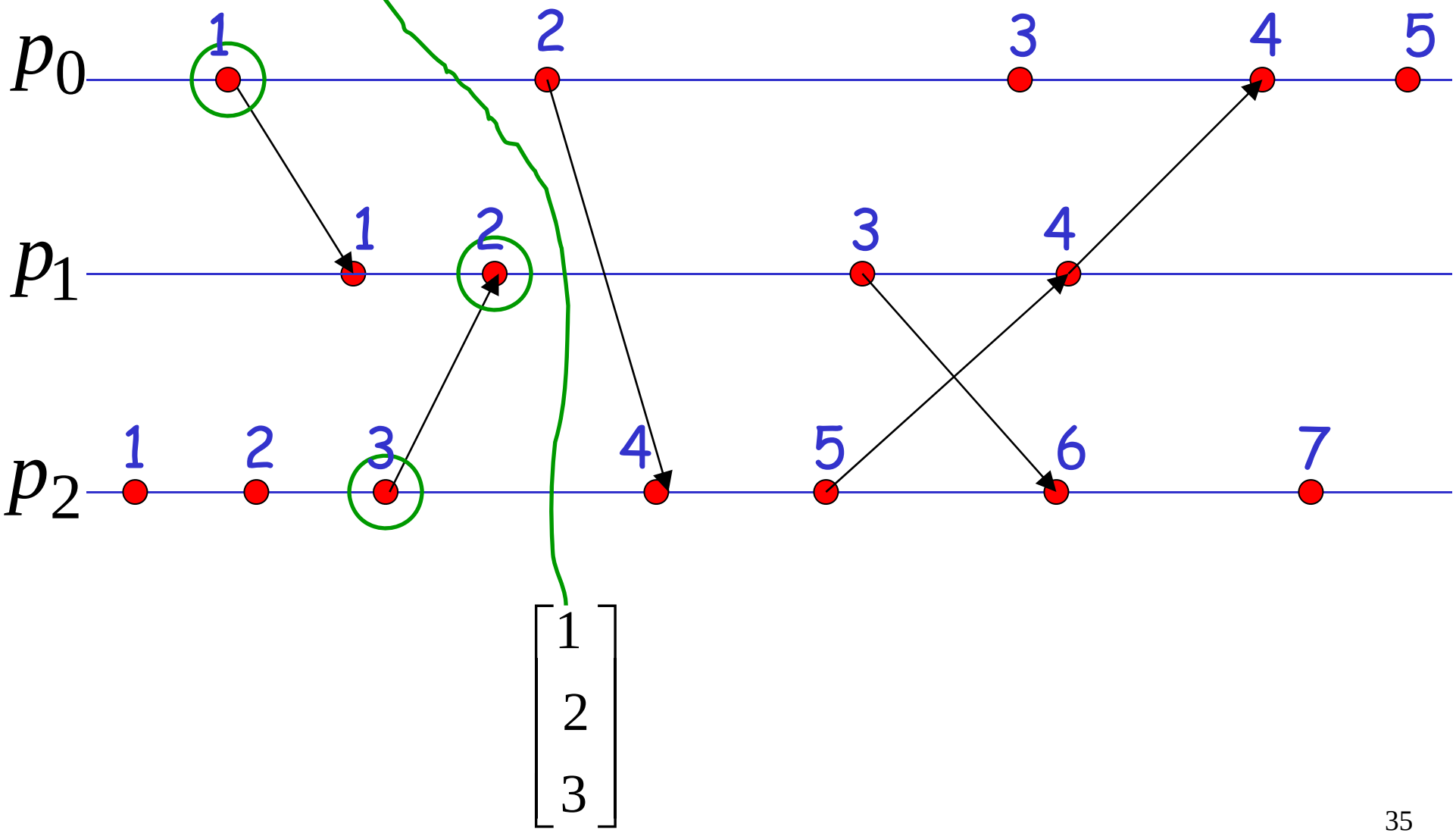
$$\varphi_1 \| \varphi_{10} \Rightarrow VC(\varphi_1) \not< VC(\varphi_{10})$$


p_0
p_1
p_3
φ_1 φ_2 φ_3 φ_4 φ_5
φ_6 φ_7 φ_8 φ_9
φ_{10} φ_{11} φ_{12} φ_{13} φ_{14} φ_{15} φ_{16}
$VC(\varphi_8) \not< VC(\varphi_{14})$
$\varphi_8 \| \varphi_{14}$

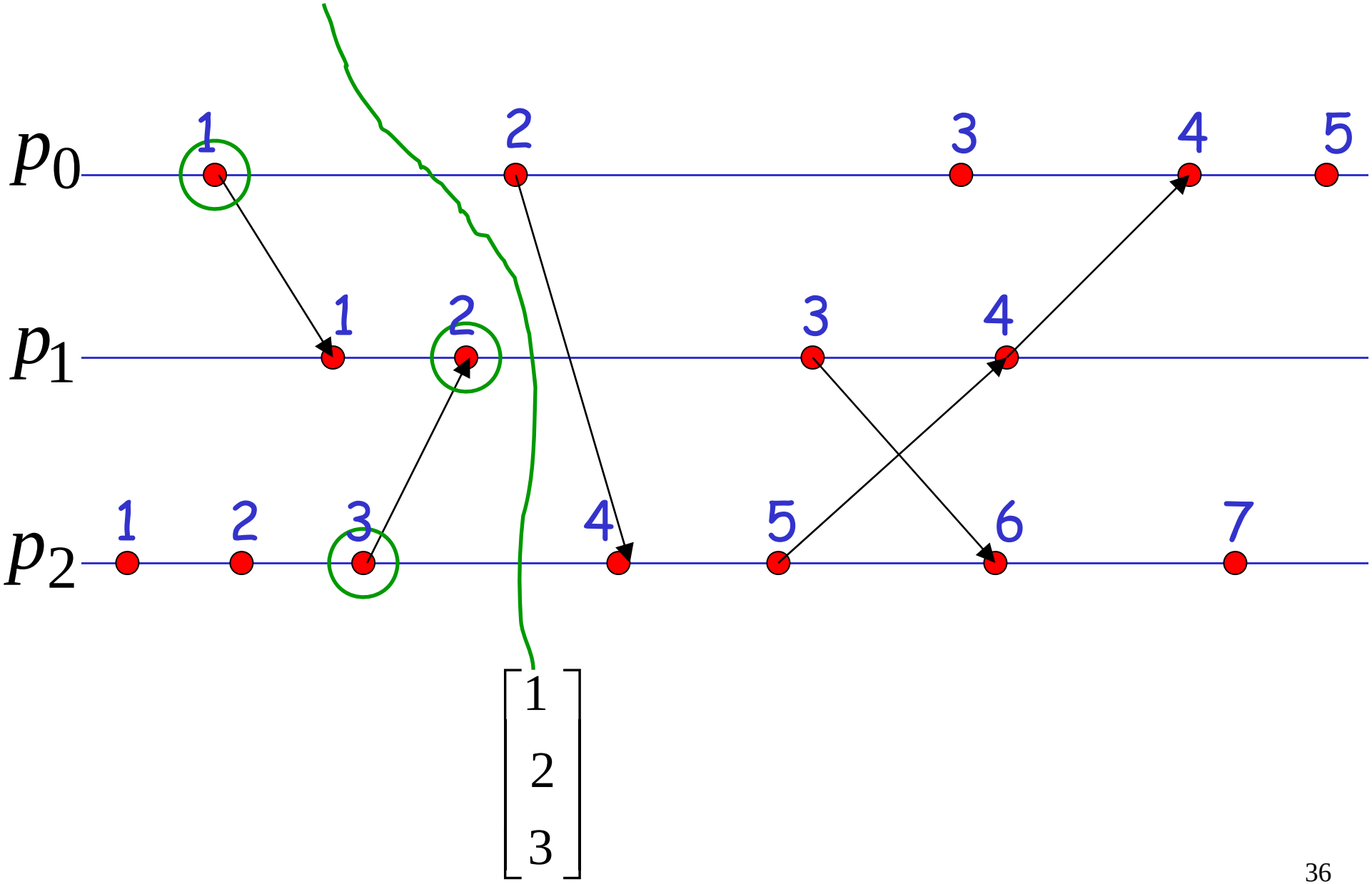
Isečci (Cuts)


p_0
p_1
p_2
1
2
3
4
5
1
2
3
4
1
2
3
4
5
6
7

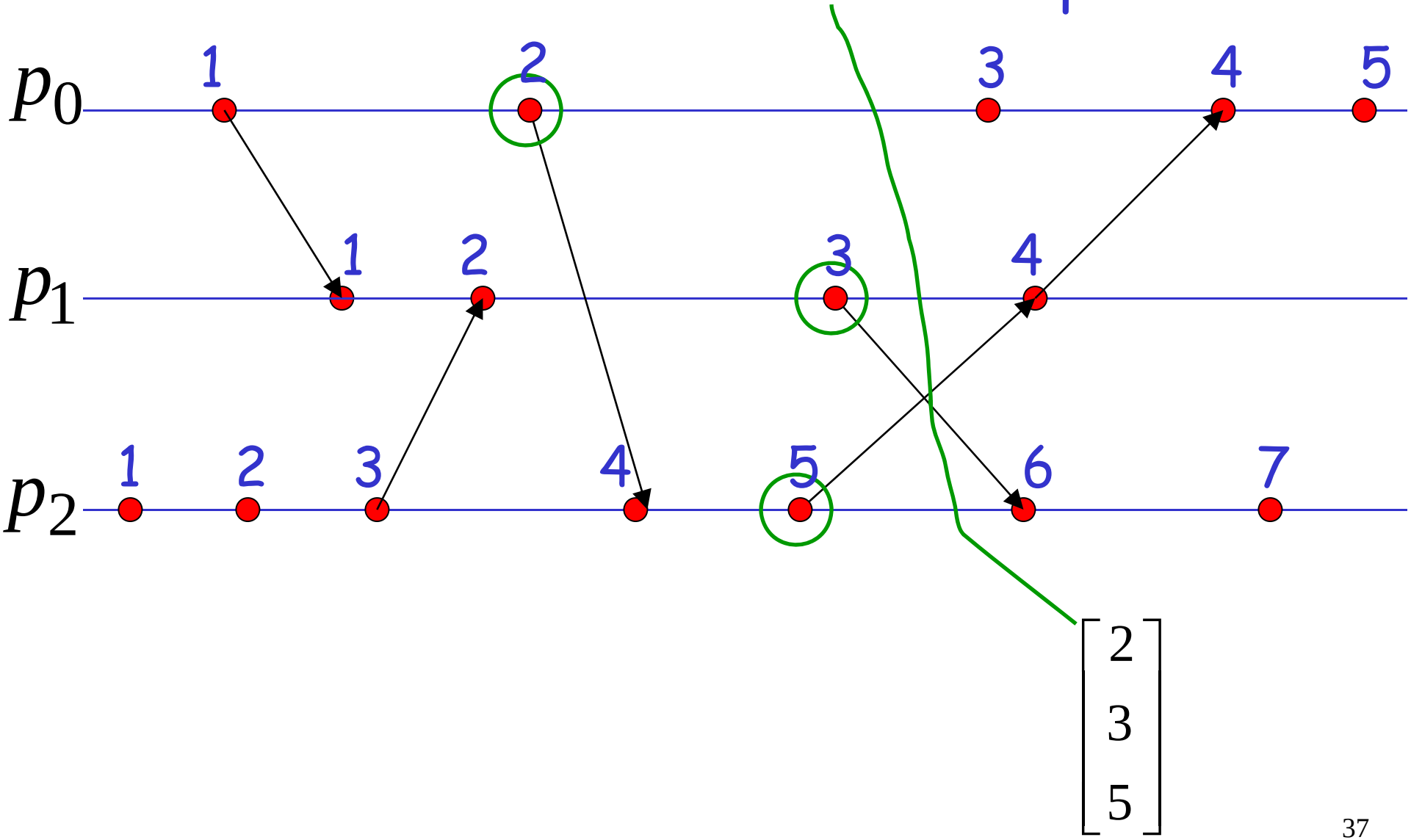
Isečak: sastoji se od događaja iz svakog procesa



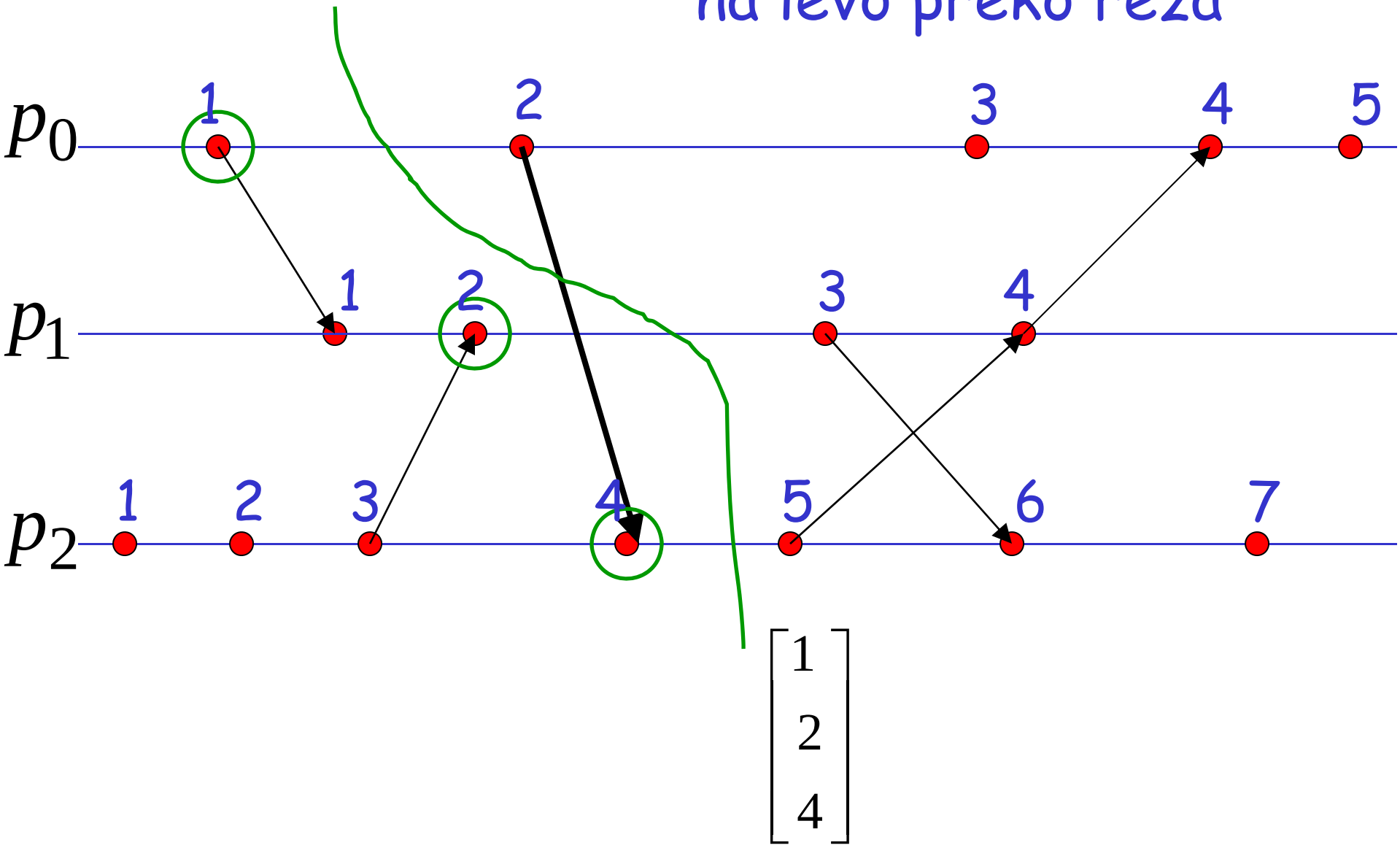
Konzistentan isečak: nema poruka preko reza



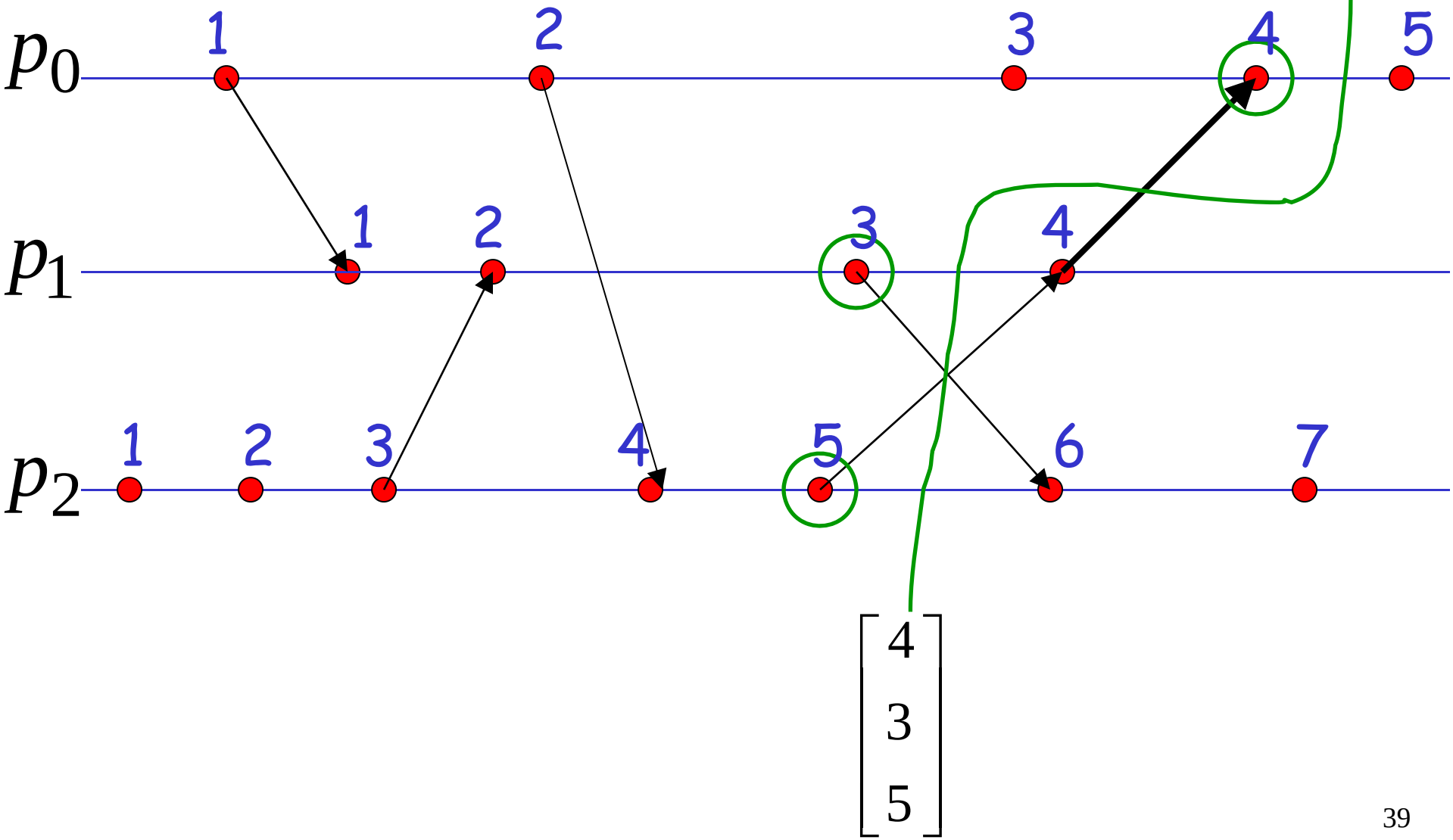
Konzistentan isečak : poruka može preći s leva u desno preko reza



Nekonzistentan isečak: poruka prelazi s desna na levo preko reza



Nekonzistentan isečak: poruka prelazi s desna na levo preko reza



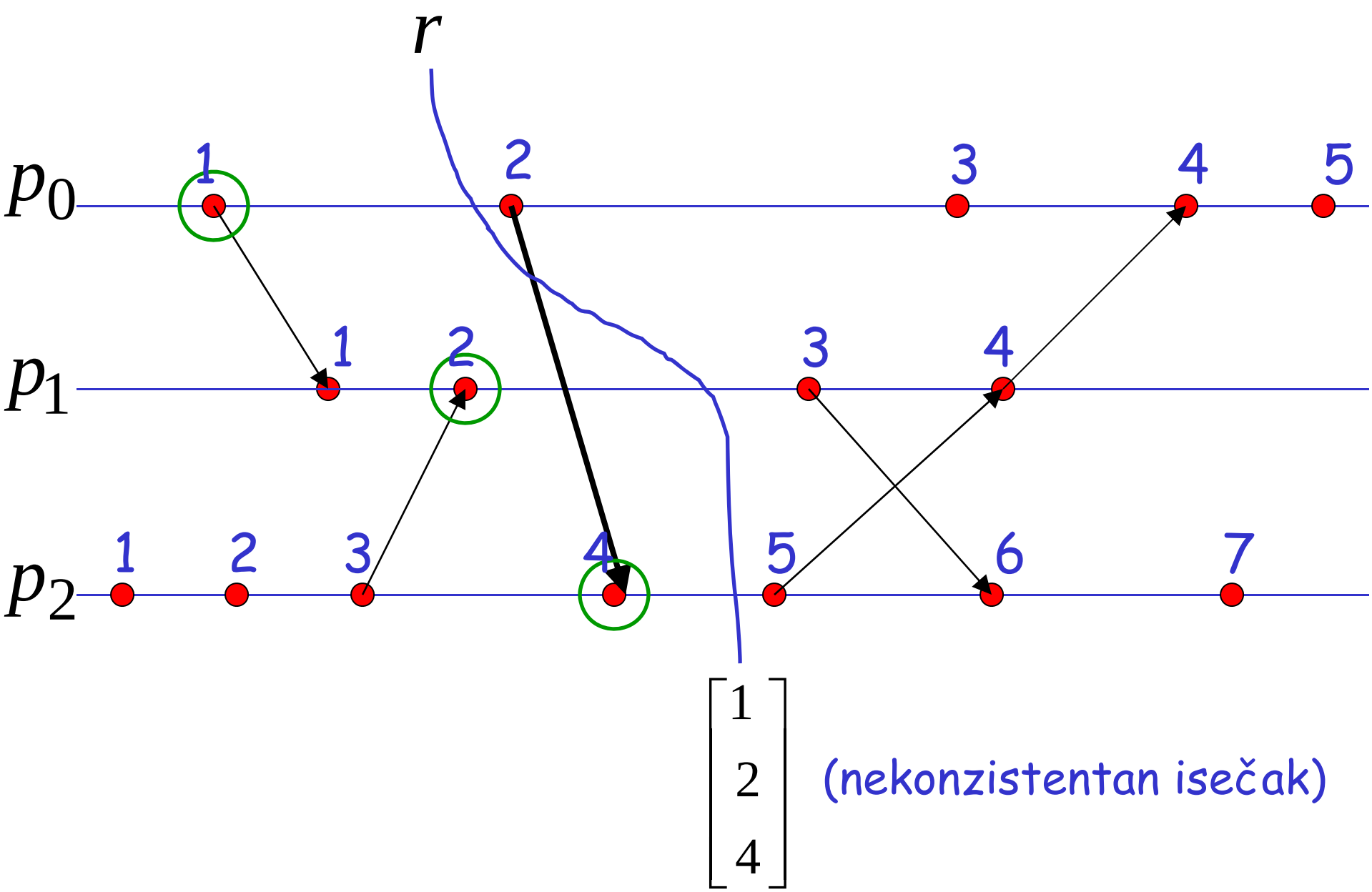
Maksimalan konzistentan isečak

Uzmimo neki (nekonzistentan) isečak r

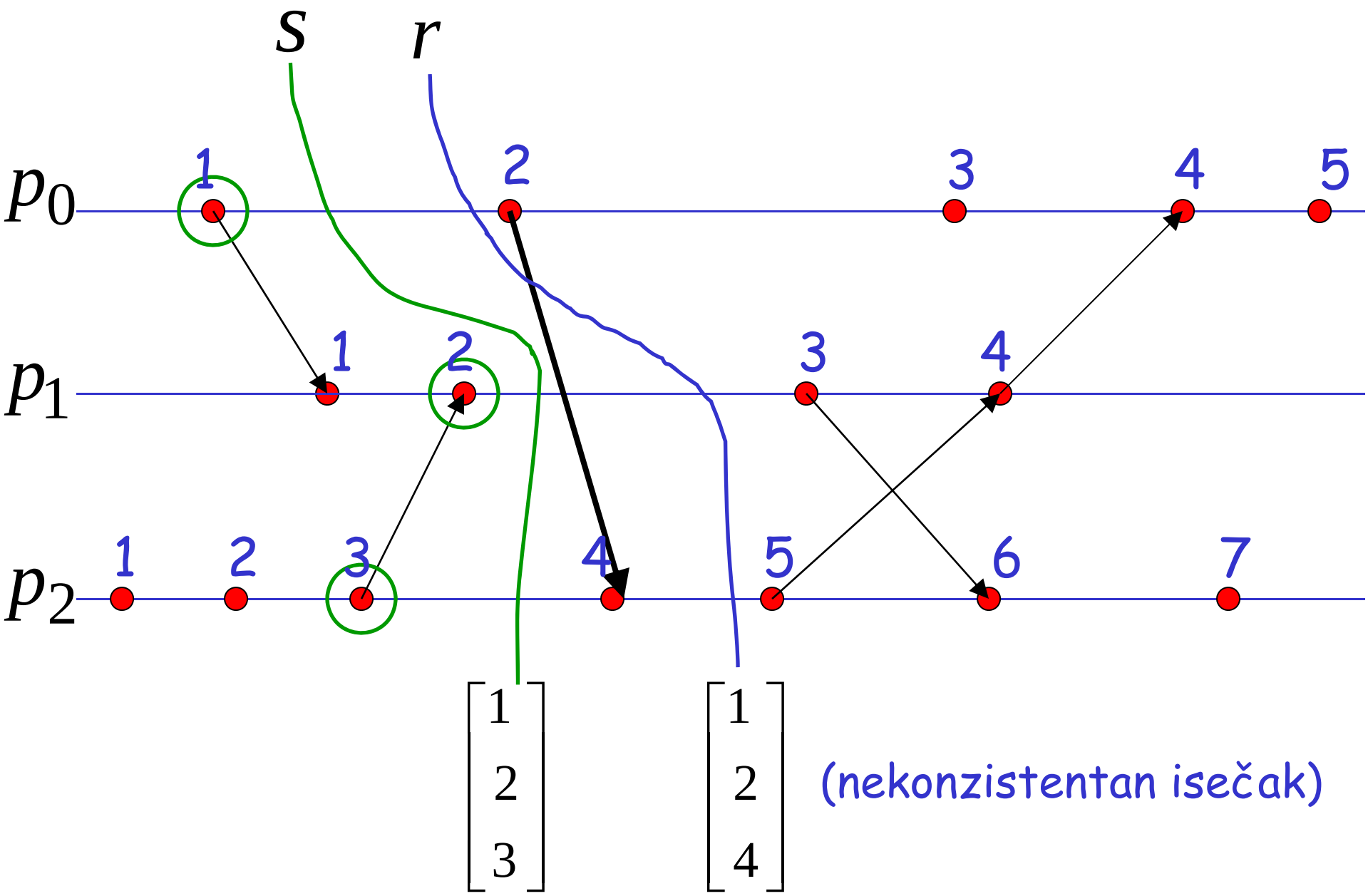
Max konzistentan isečak od: r

Neki konzistentan isečak s takav da je $s \leq r$

i s sadrži najnovije događaje


r
p_0
p_1
p_2
1
2
3
4
5
1
2
3
4
1
2
3
4
5
6
7
$\begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$
(nekonzistentan isečak)

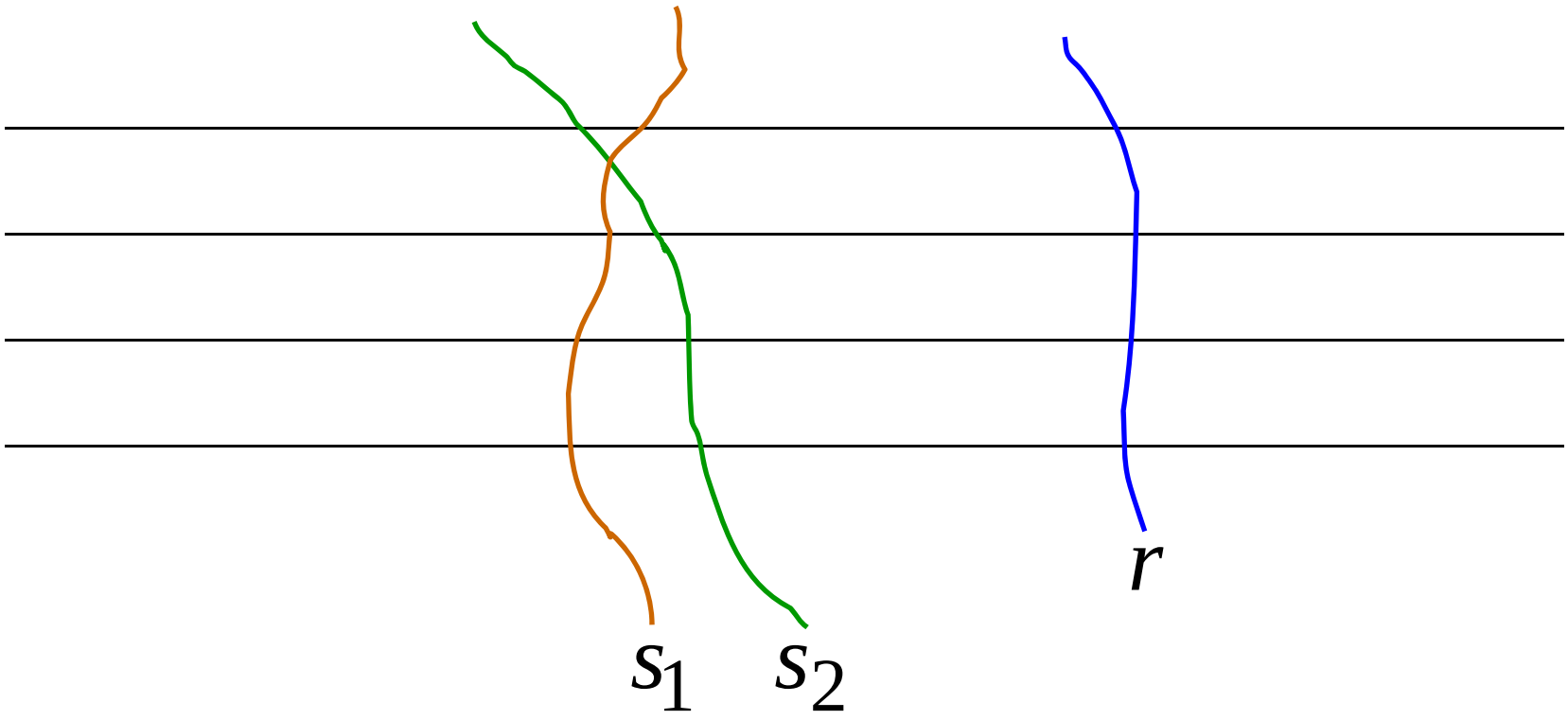
maksimalan konzistentan isečak



Teorema: Za svaki isečak r,
postoji jedinstven
max konzistentan isečak

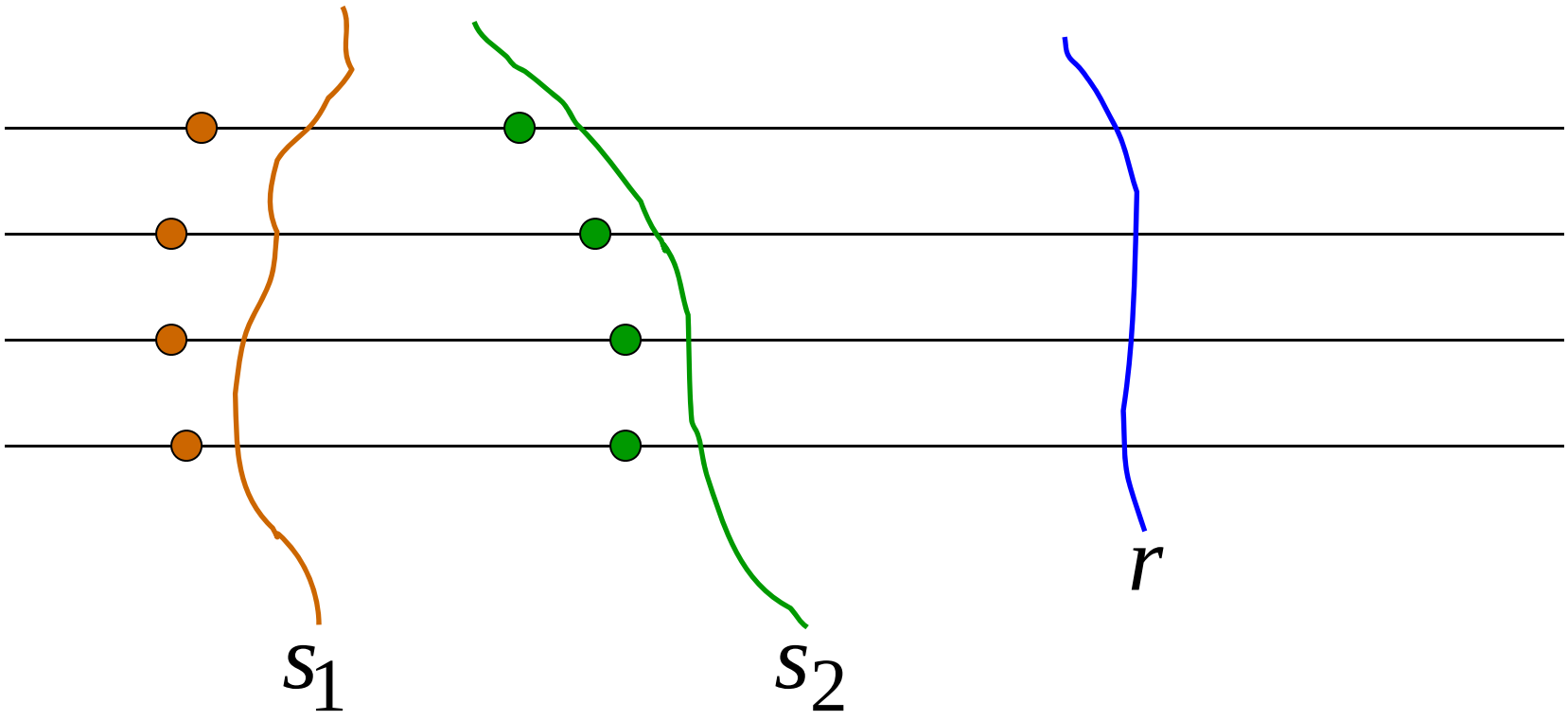
Dokaz: Dokaz kontradikcijom

Pret. radi kontradikcije da ima dva (ili više) max konz. isečaka od r



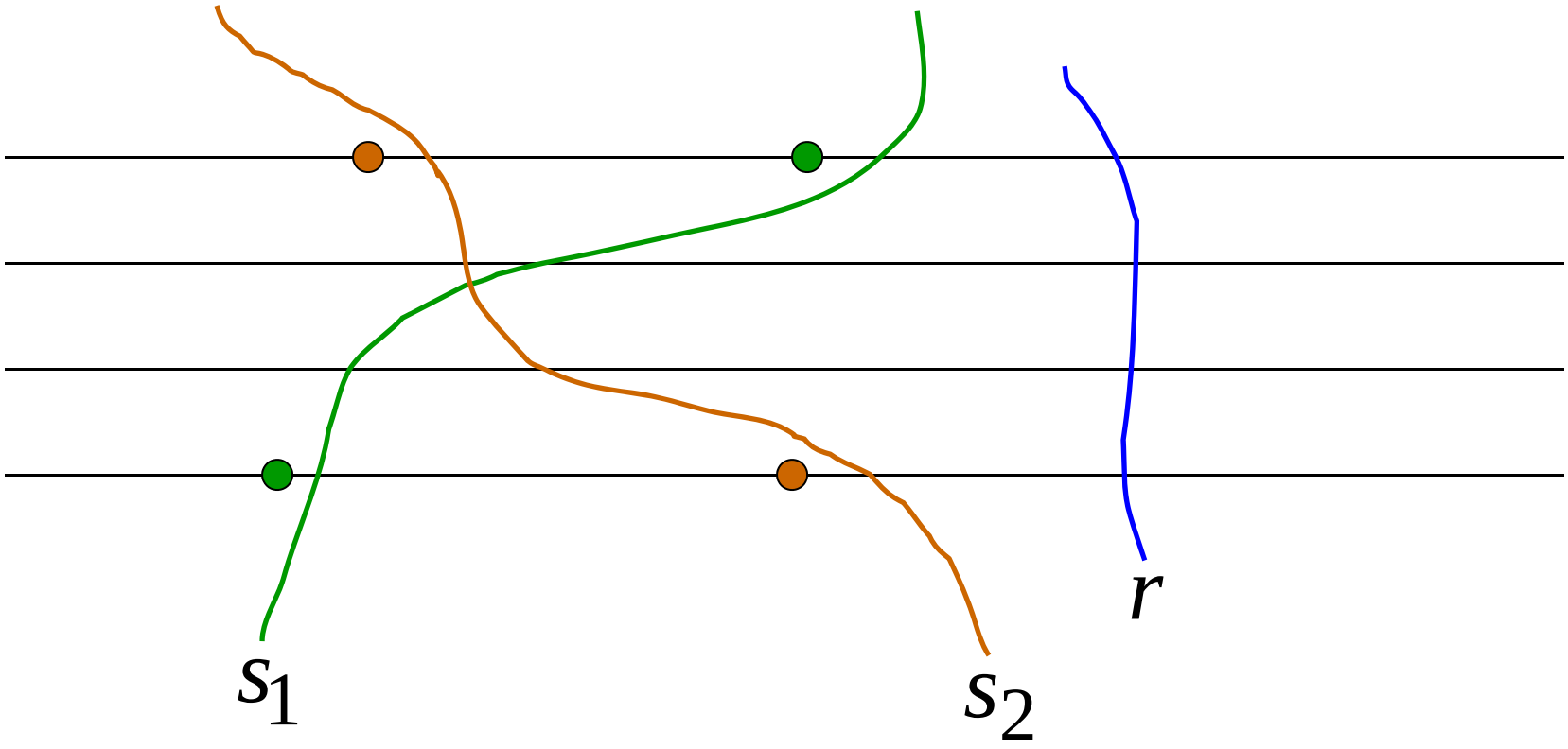
maksimalni konzistentni isečci

Nemože biti da se s_1 i s_2 ne ukrštaju

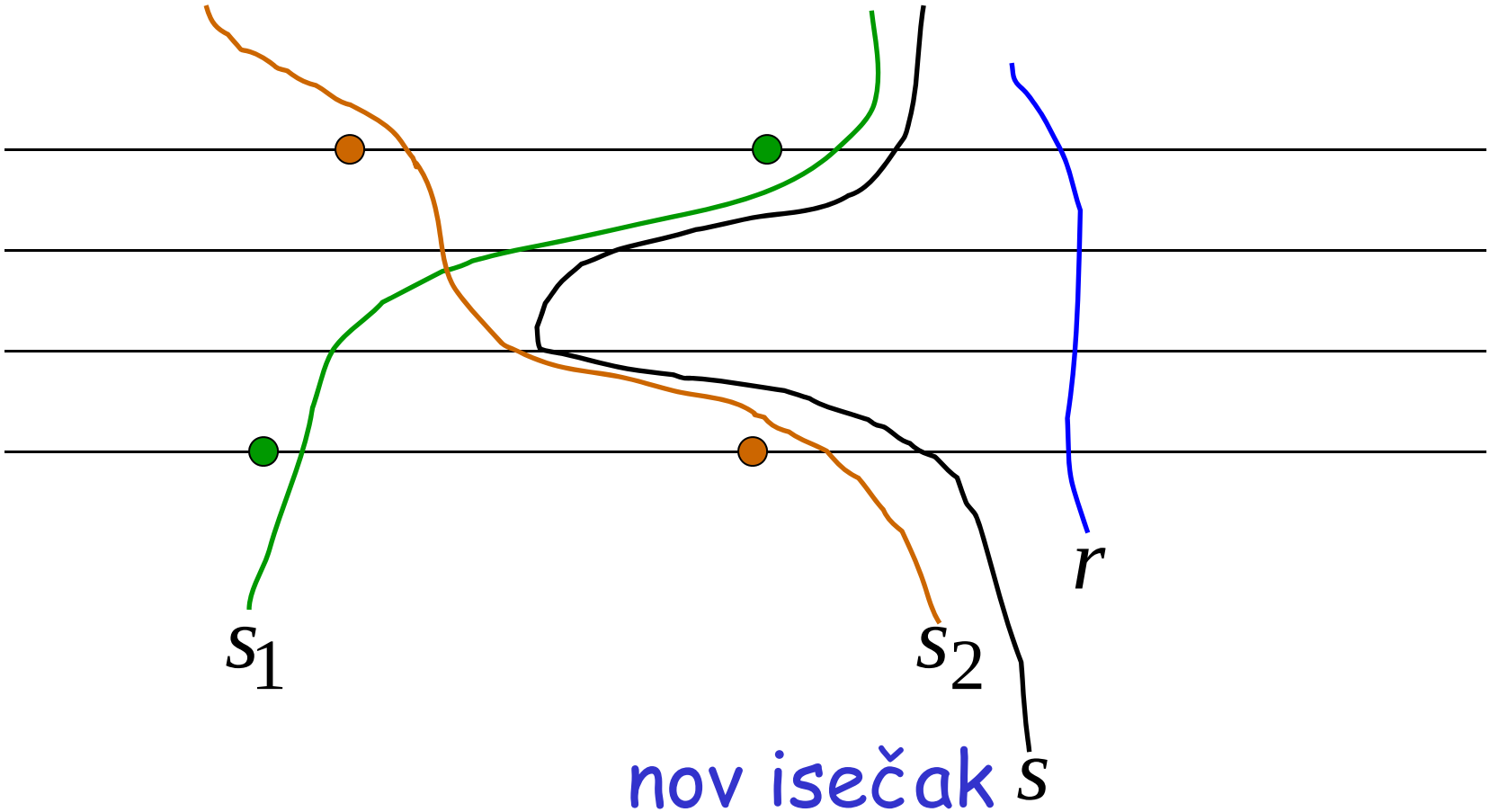


nije maksimalan!

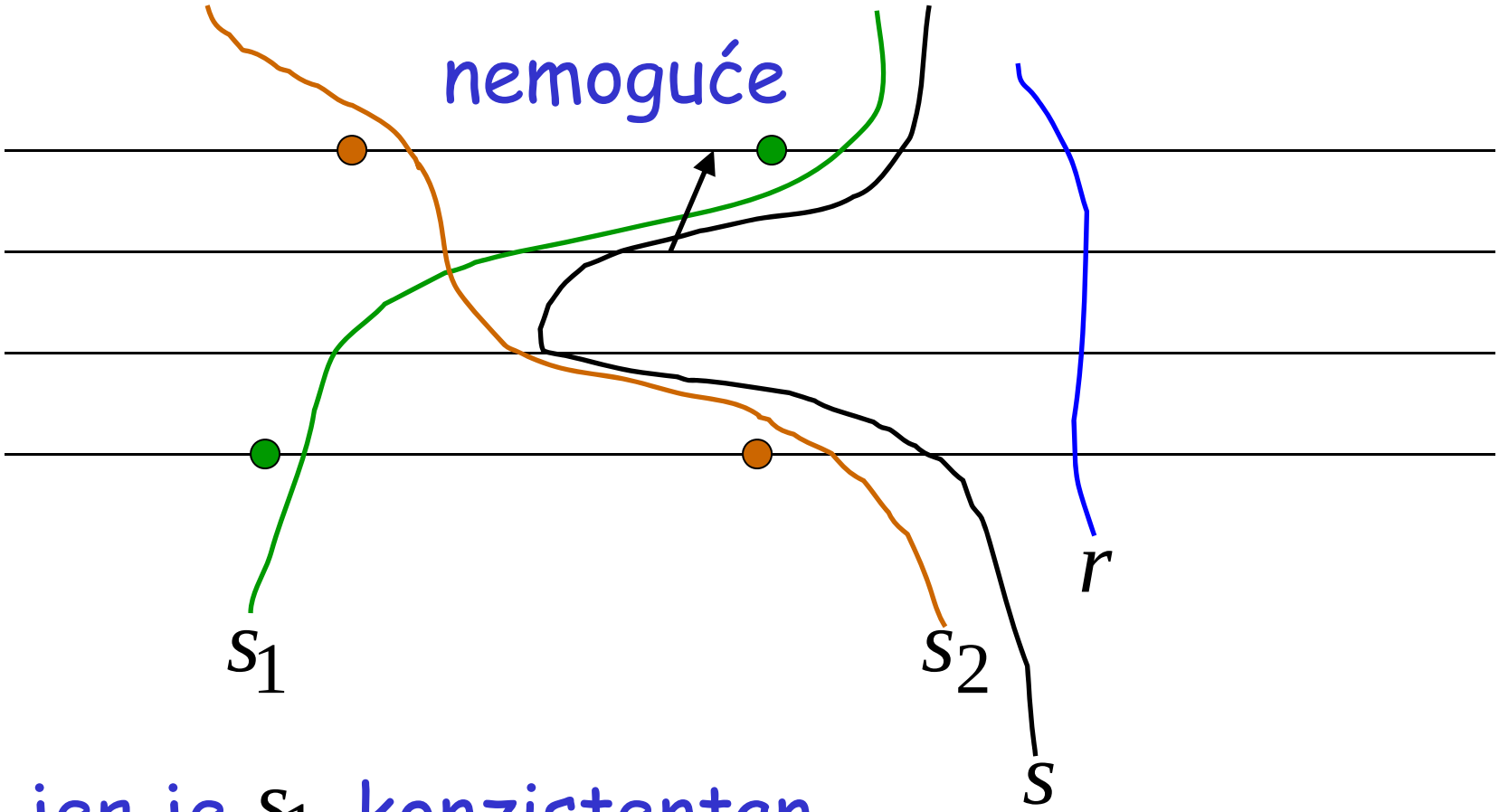
s_1 i s_2 se ukrštaju



s_1 i s_2 se ukrštaju

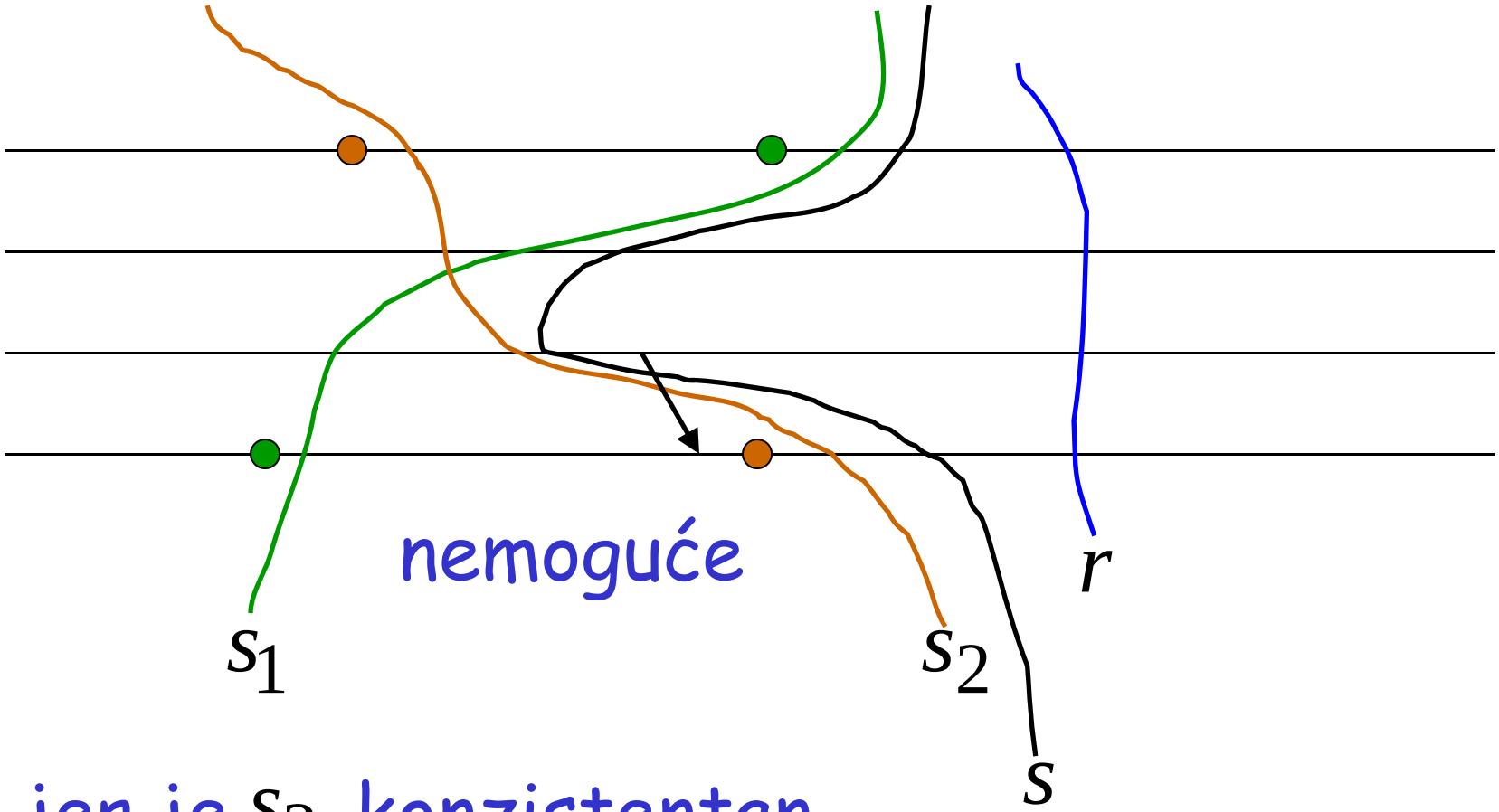


s_1 i s_2 se ukrštaju



jer je s_1 konzistentan

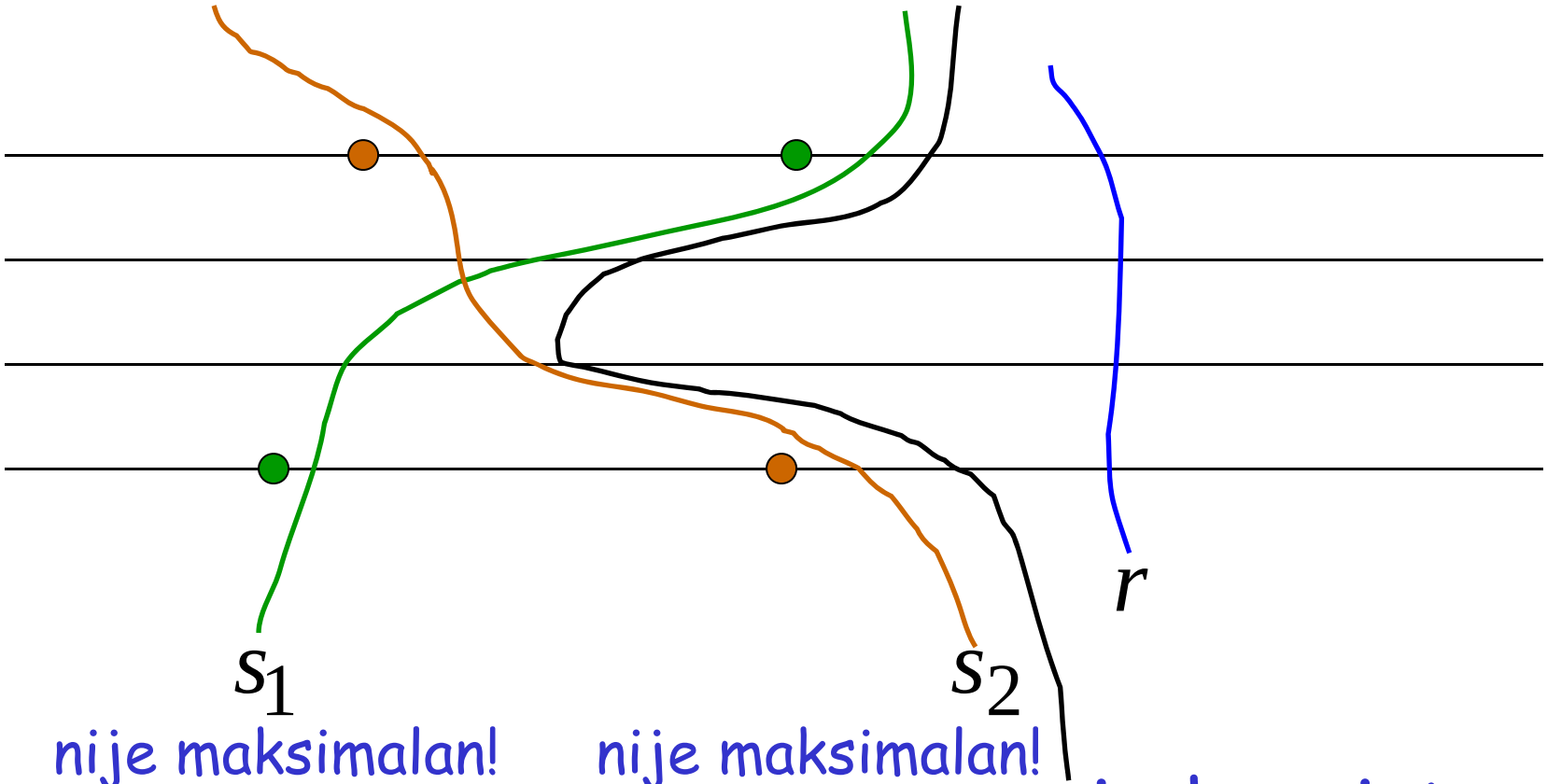
s_1 i s_2 se ukrštaju



jer je s_2 konzistentan

s_1 i s_2 se ukrštaju

Kontradikcija!



s_1 nije maksimalan!

s_2 nije maksimalan!

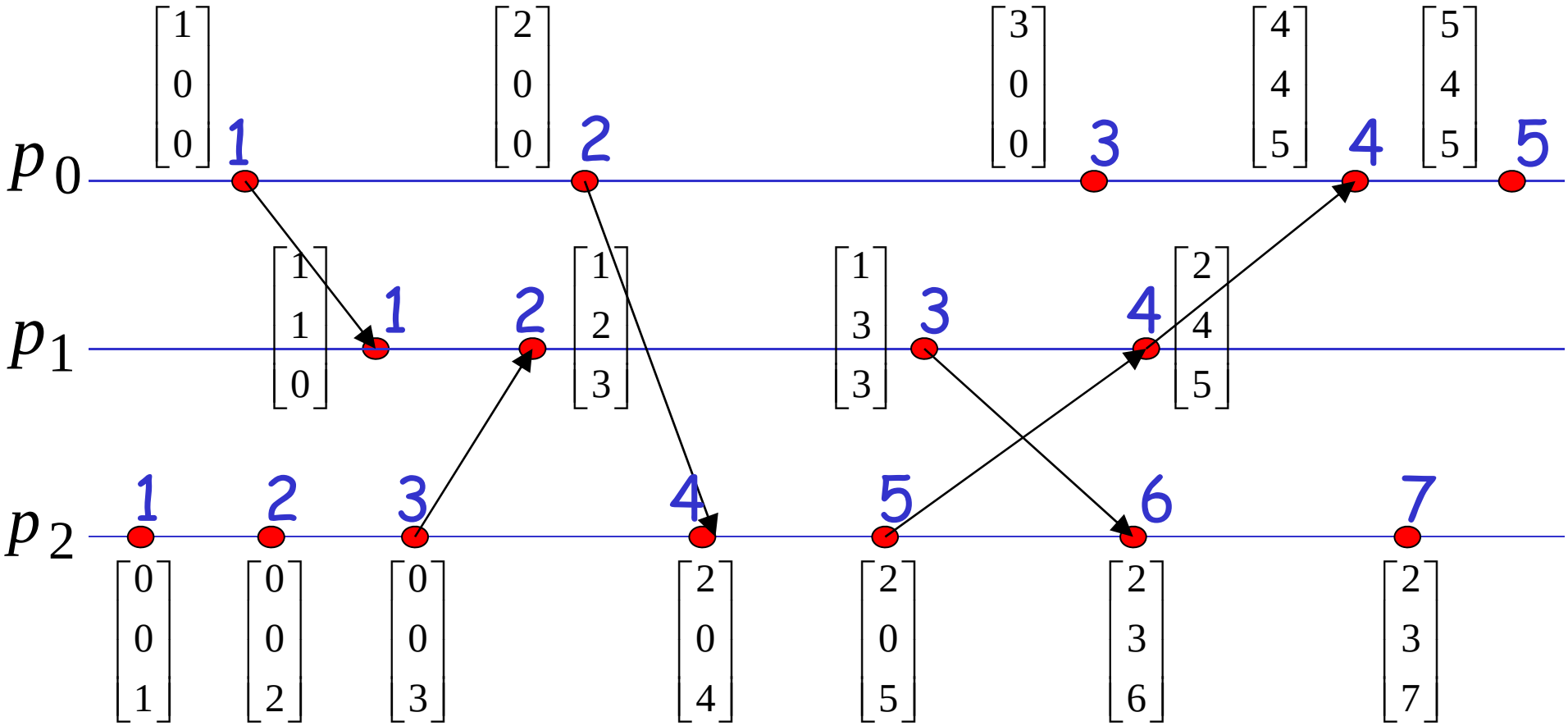
s je konzistentan

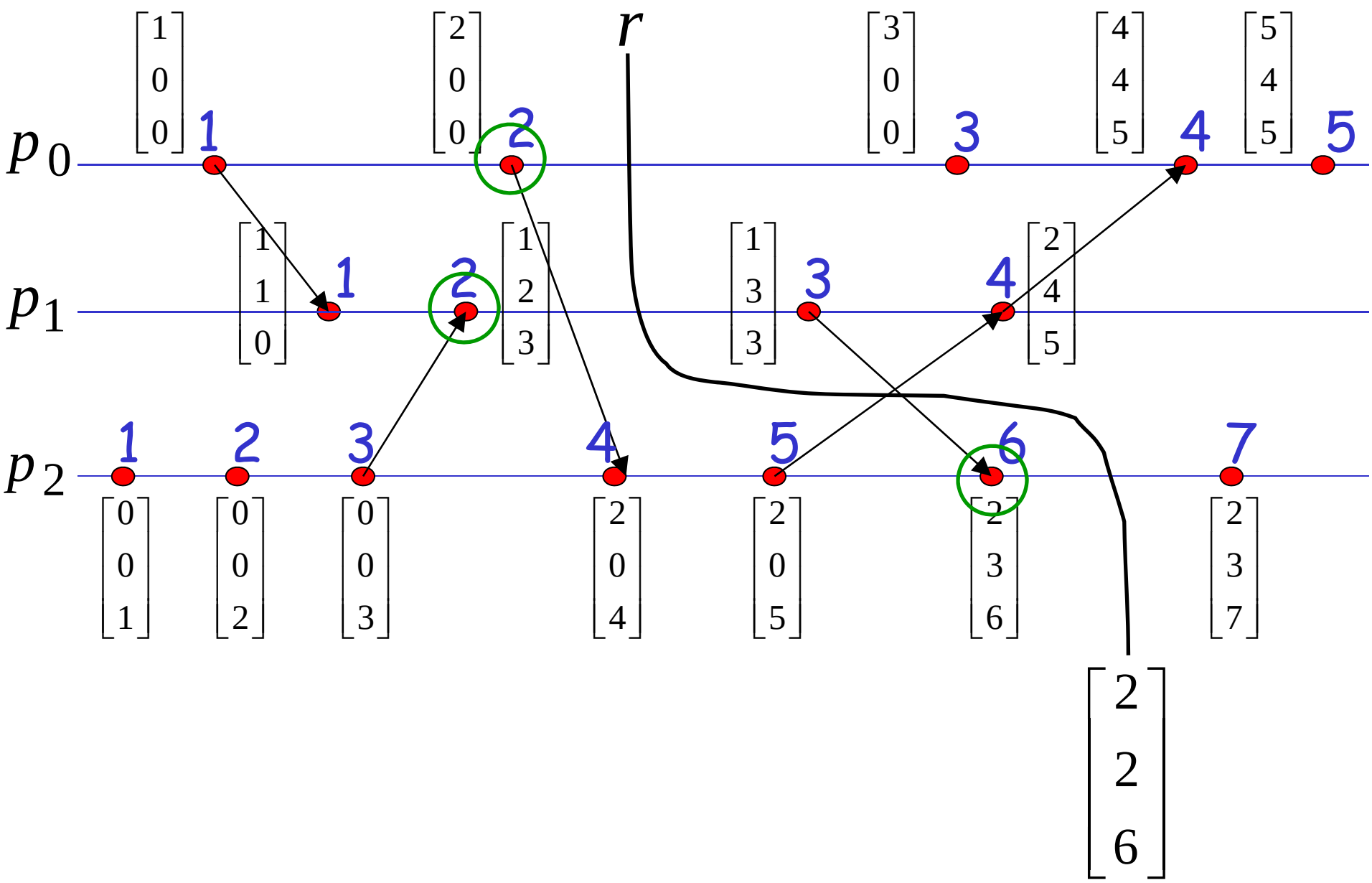
Kraj dokaza

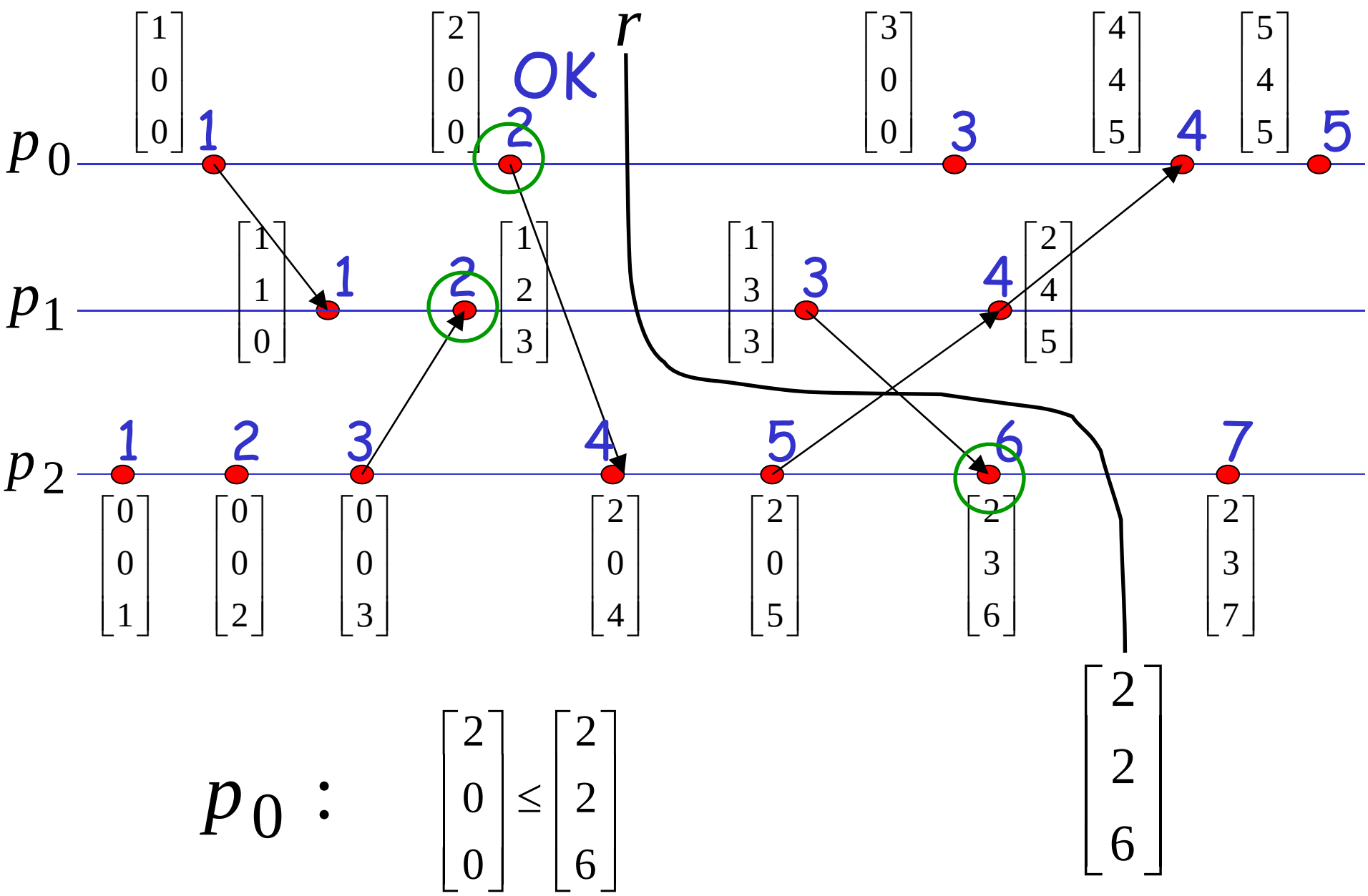
Distribuirani algoritam za računanje maksimalnog konzistentnog isečka od: r

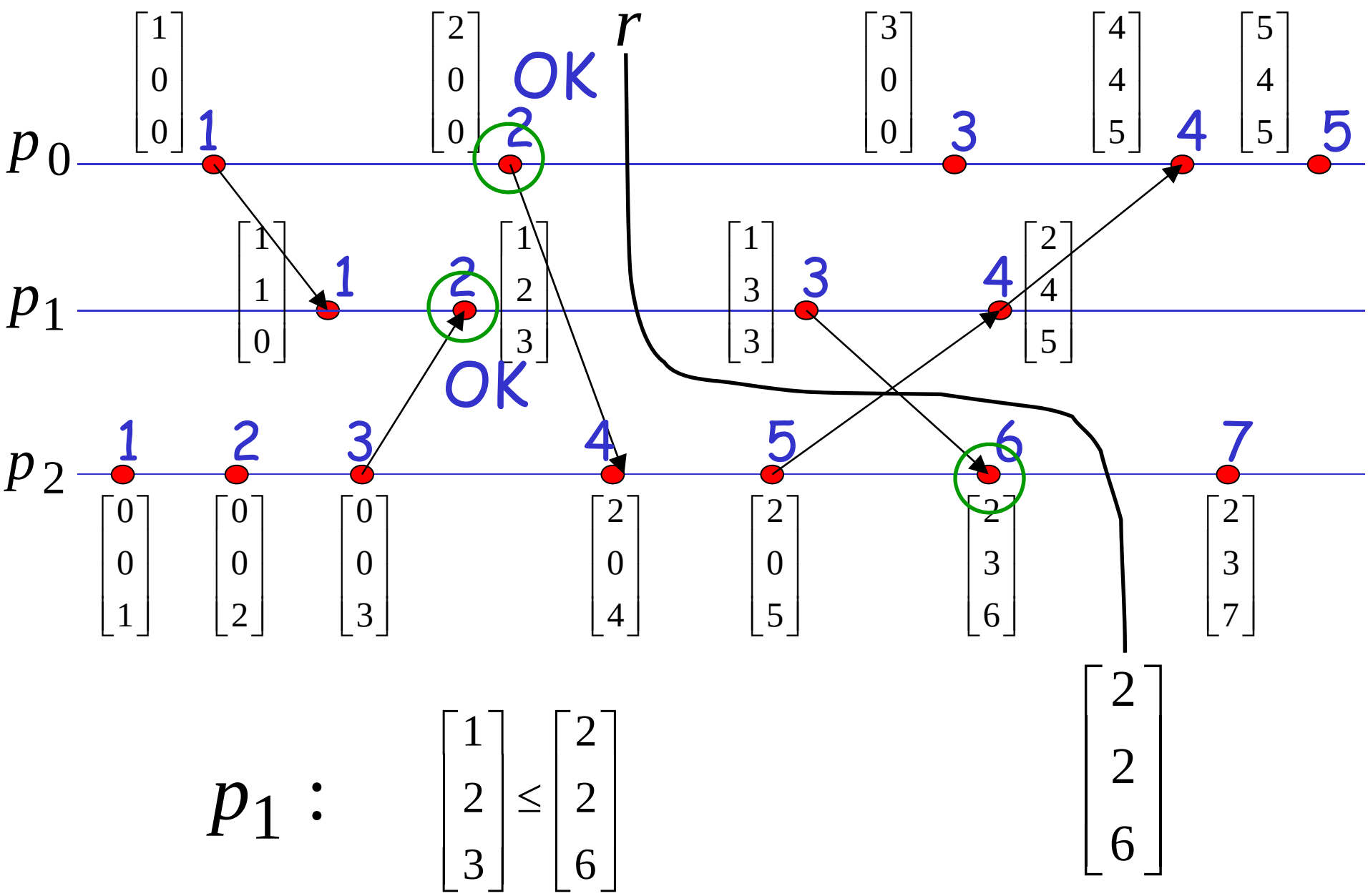
- Koristi vektorske satove
- Za svaki procesor:

  Nađi najnoviji događaj sa vektorskim satom $v \leq r$

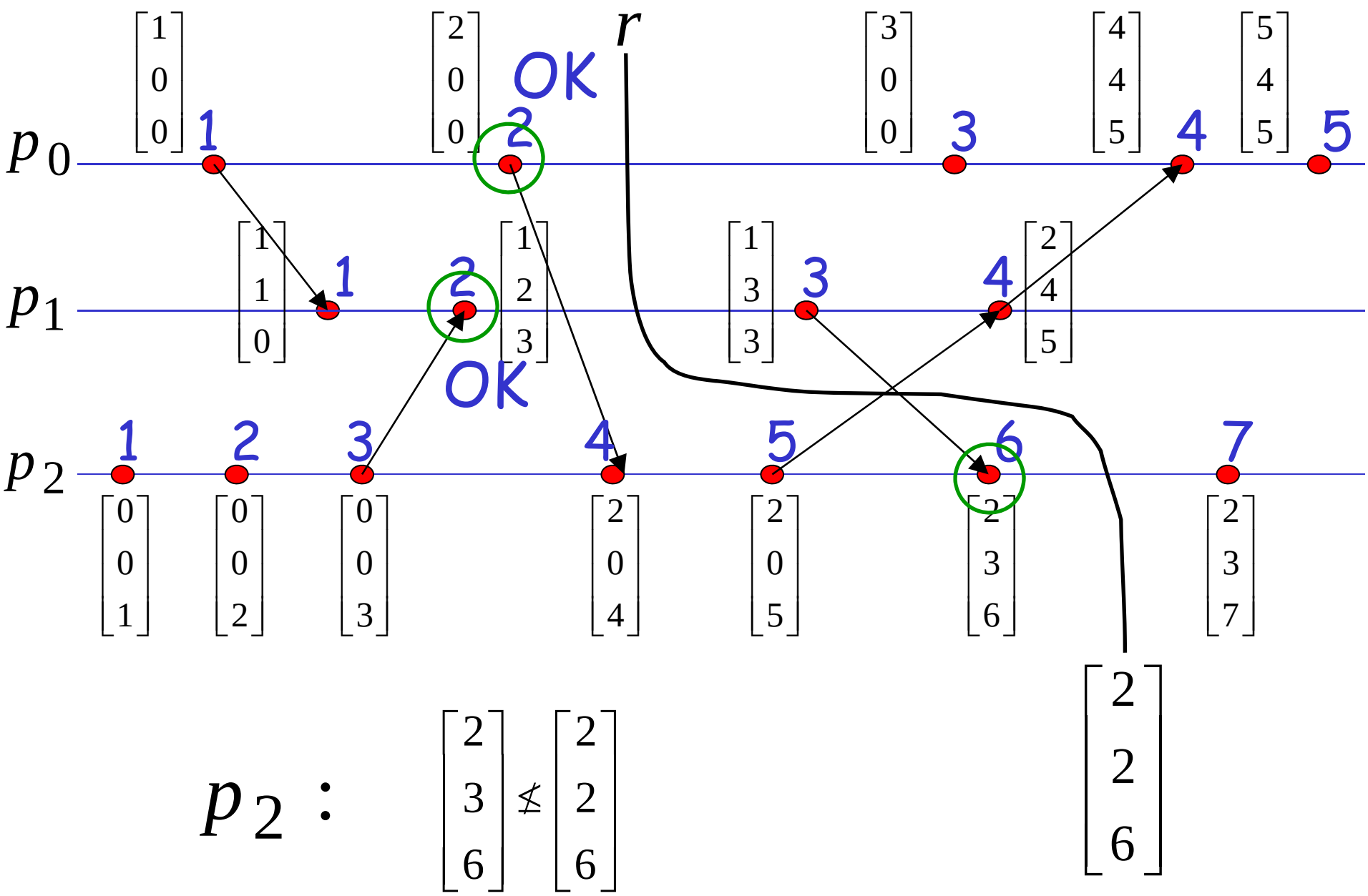

p_0
p_1
p_2
1
2
3
4
5
6
7
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}2\\0\\0\end{bmatrix}$
$\begin{bmatrix}3\\0\\0\end{bmatrix}$
$\begin{bmatrix}4\\4\\5\end{bmatrix}$
$\begin{bmatrix}5\\4\\5\end{bmatrix}$
$\begin{bmatrix}1\\1\\0\end{bmatrix}$
$\begin{bmatrix}1\\2\\3\end{bmatrix}$
$\begin{bmatrix}1\\3\\3\end{bmatrix}$
$\begin{bmatrix}2\\4\\5\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\2\end{bmatrix}$
$\begin{bmatrix}0\\0\\3\end{bmatrix}$
$\begin{bmatrix}2\\0\\4\end{bmatrix}$
$\begin{bmatrix}2\\0\\5\end{bmatrix}$
$\begin{bmatrix}2\\3\\6\end{bmatrix}$
$\begin{bmatrix}2\\3\\7\end{bmatrix}$

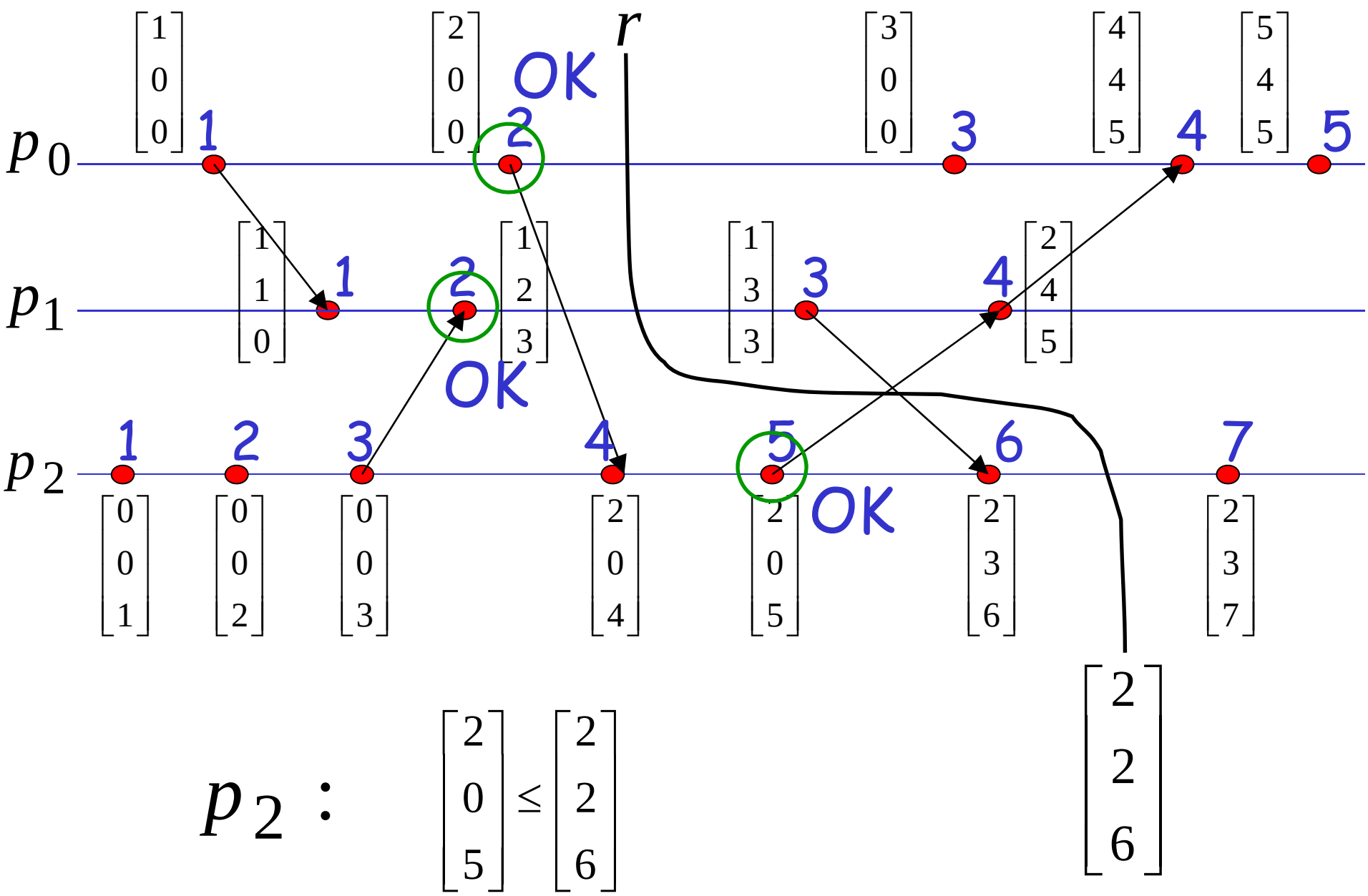

p_0
p_1
p_2
r
1 2 3 4 5
1 2 3 4
1 2 3 4 5 6 7
$\begin{bmatrix}1\\0\\0\end{bmatrix}$ $\begin{bmatrix}2\\0\\0\end{bmatrix}$ $\begin{bmatrix}3\\0\\0\end{bmatrix}$ $\begin{bmatrix}4\\4\\5\end{bmatrix}$ $\begin{bmatrix}5\\4\\5\end{bmatrix}$
$\begin{bmatrix}1\\1\\0\end{bmatrix}$ $\begin{bmatrix}1\\2\\3\end{bmatrix}$ $\begin{bmatrix}1\\3\\3\end{bmatrix}$ $\begin{bmatrix}2\\4\\5\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$ $\begin{bmatrix}0\\0\\2\end{bmatrix}$ $\begin{bmatrix}0\\0\\3\end{bmatrix}$ $\begin{bmatrix}2\\0\\4\end{bmatrix}$ $\begin{bmatrix}2\\0\\5\end{bmatrix}$ $\begin{bmatrix}2\\3\\6\end{bmatrix}$ $\begin{bmatrix}2\\3\\7\end{bmatrix}$
$\begin{bmatrix}2\\2\\6\end{bmatrix}$

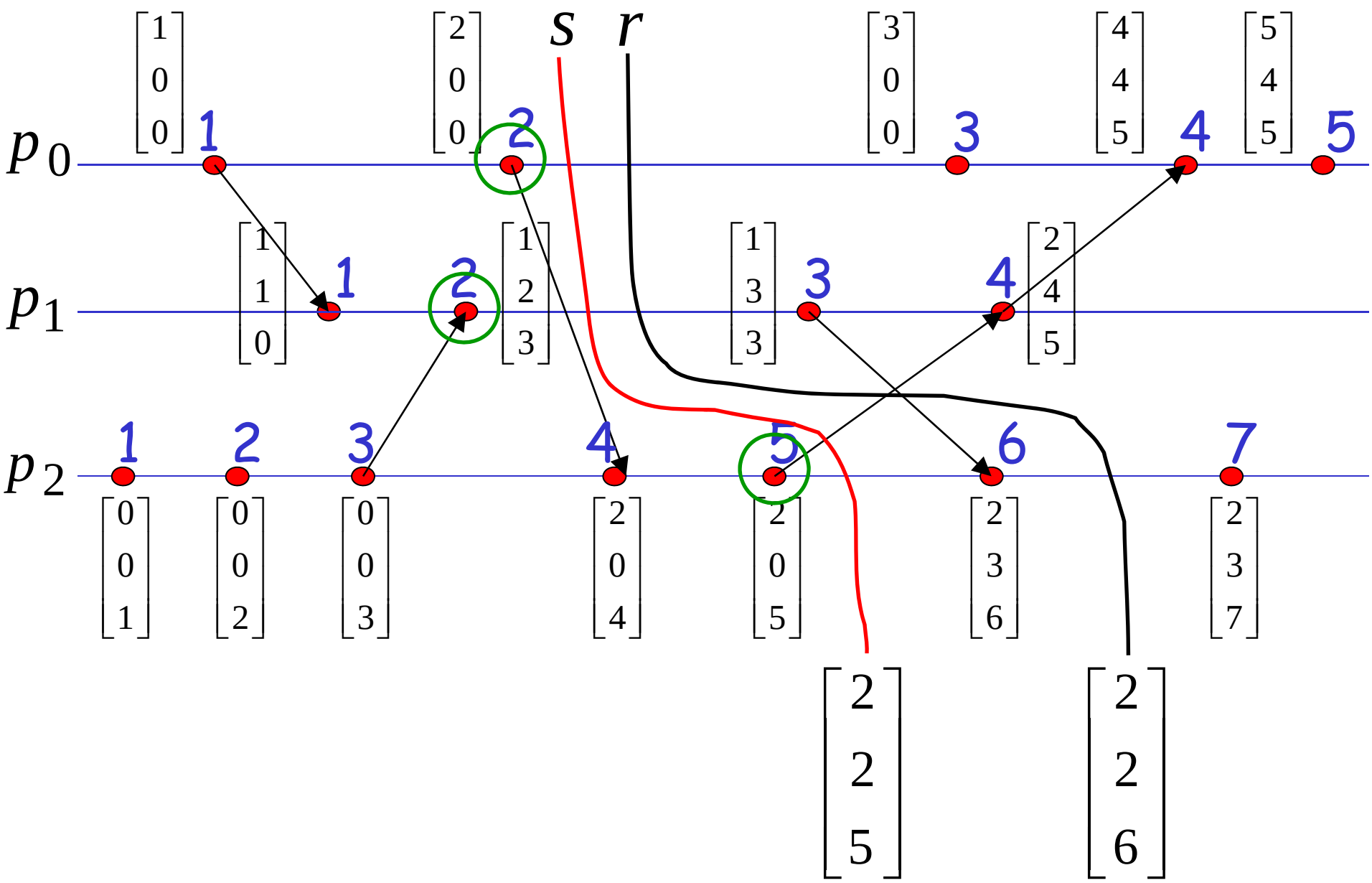

r
OK
p_0
p_1
p_2
1
2
3
4
5
6
7
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}2\\0\\0\end{bmatrix}$
$\begin{bmatrix}3\\0\\0\end{bmatrix}$
$\begin{bmatrix}4\\4\\5\end{bmatrix}$
$\begin{bmatrix}5\\4\\5\end{bmatrix}$
$\begin{bmatrix}1\\1\\0\end{bmatrix}$
$\begin{bmatrix}1\\2\\3\end{bmatrix}$
$\begin{bmatrix}1\\3\\3\end{bmatrix}$
$\begin{bmatrix}2\\4\\5\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\2\end{bmatrix}$
$\begin{bmatrix}0\\0\\3\end{bmatrix}$
$\begin{bmatrix}2\\0\\4\end{bmatrix}$
$\begin{bmatrix}2\\0\\5\end{bmatrix}$
$\begin{bmatrix}2\\3\\6\end{bmatrix}$
$\begin{bmatrix}2\\3\\7\end{bmatrix}$
$p_0 : \begin{bmatrix}2\\0\\0\end{bmatrix} \leq \begin{bmatrix}2\\2\\6\end{bmatrix}$
$\begin{bmatrix}2\\2\\6\end{bmatrix}$



$p_1 : \begin{bmatrix}1\\2\\3\end{bmatrix} \leq \begin{bmatrix}2\\2\\6\end{bmatrix}$


p_0
p_1
p_2
r
OK
OK
$\begin{bmatrix}1\\0\\0\end{bmatrix}$ $\begin{bmatrix}2\\0\\0\end{bmatrix}$ $\begin{bmatrix}3\\0\\0\end{bmatrix}$ $\begin{bmatrix}4\\4\\5\end{bmatrix}$ $\begin{bmatrix}5\\4\\5\end{bmatrix}$
$\begin{bmatrix}1\\1\\0\end{bmatrix}$ $\begin{bmatrix}1\\2\\3\end{bmatrix}$ $\begin{bmatrix}1\\3\\3\end{bmatrix}$ $\begin{bmatrix}2\\4\\5\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$ $\begin{bmatrix}0\\0\\2\end{bmatrix}$ $\begin{bmatrix}0\\0\\3\end{bmatrix}$ $\begin{bmatrix}2\\0\\4\end{bmatrix}$ $\begin{bmatrix}2\\0\\5\end{bmatrix}$ $\begin{bmatrix}2\\3\\6\end{bmatrix}$ $\begin{bmatrix}2\\3\\7\end{bmatrix}$
$\begin{bmatrix}2\\2\\6\end{bmatrix}$
p_2 : $\begin{bmatrix}2\\3\\6\end{bmatrix} \not\leq \begin{bmatrix}2\\2\\6\end{bmatrix}$


p_0
p_1
p_2
r
OK
OK
OK
p_2 : $\begin{bmatrix}2\\0\\5\end{bmatrix} \le \begin{bmatrix}2\\2\\6\end{bmatrix}$


p_0
p_1
p_2
s
r
1
2
3
4
5
6
7
[1 0 0]
[2 0 0]
[3 0 0]
[4 4 5]
[5 4 5]
[1 1 0]
[1 2 3]
[1 3 3]
[2 4 5]
[0 0 1]
[0 0 2]
[0 0 3]
[2 0 4]
[2 0 5]
[2 3 6]
[2 3 7]
[2 2 5]
[2 2 6]

Distribuirani snimak (snapshot)

- Skup procesora S inicira računanje radi dobijanja globalnog snimka

  (ovi procesori primaju iz sistema specijalnu poruku marker)

- Isečak sadrži stanje barem jednog procesora u S u inicijalizaciji

Distribuirani algoritam snimka

Procesor: p_i $num_i = 0$ $ans_i = nil$

- Broj lokalne događaje u num_i

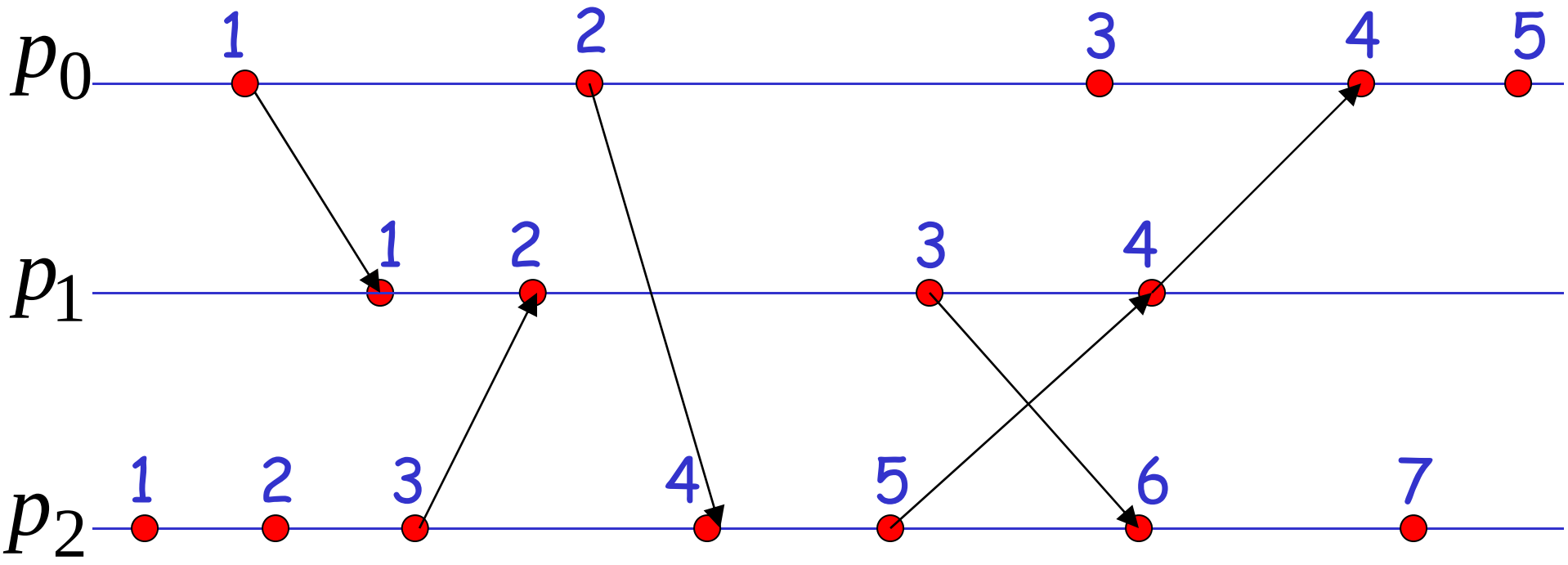
- Po prijemu poruke marker :

If $ans_i = nil$ then

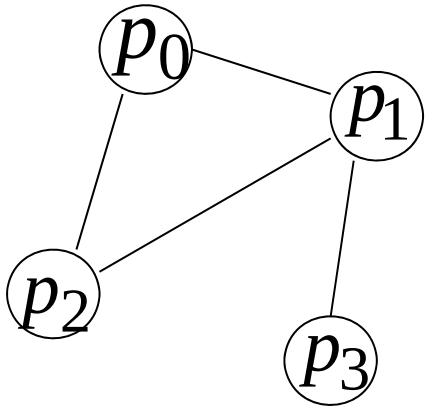
postavi $ans_i = num_i$

pošalji marker svim susedima

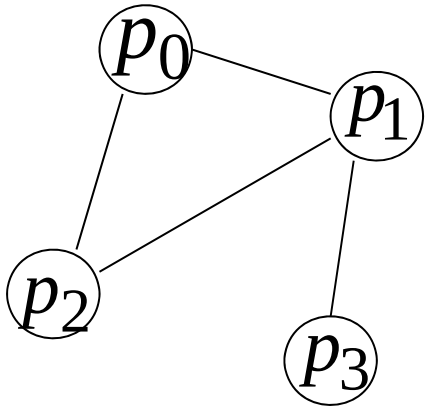
num_i



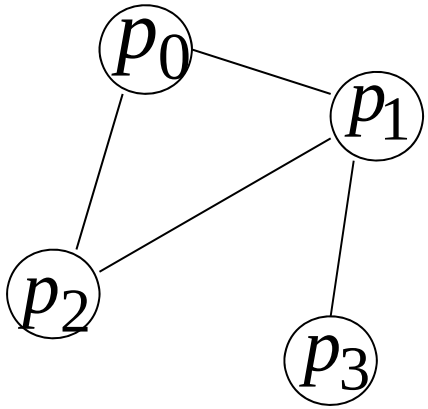
$$S = \{p_0\}$$

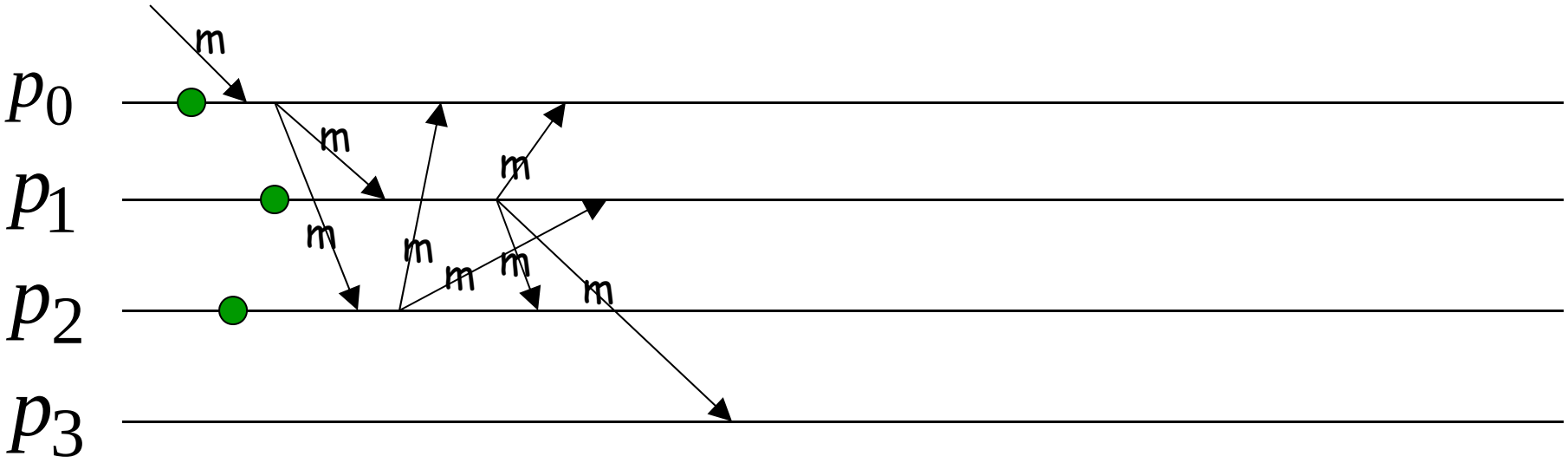


$$S = \{p_0\}$$

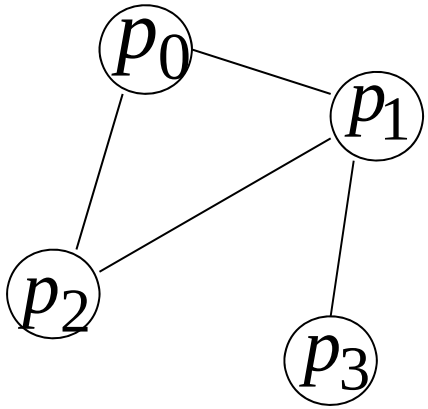


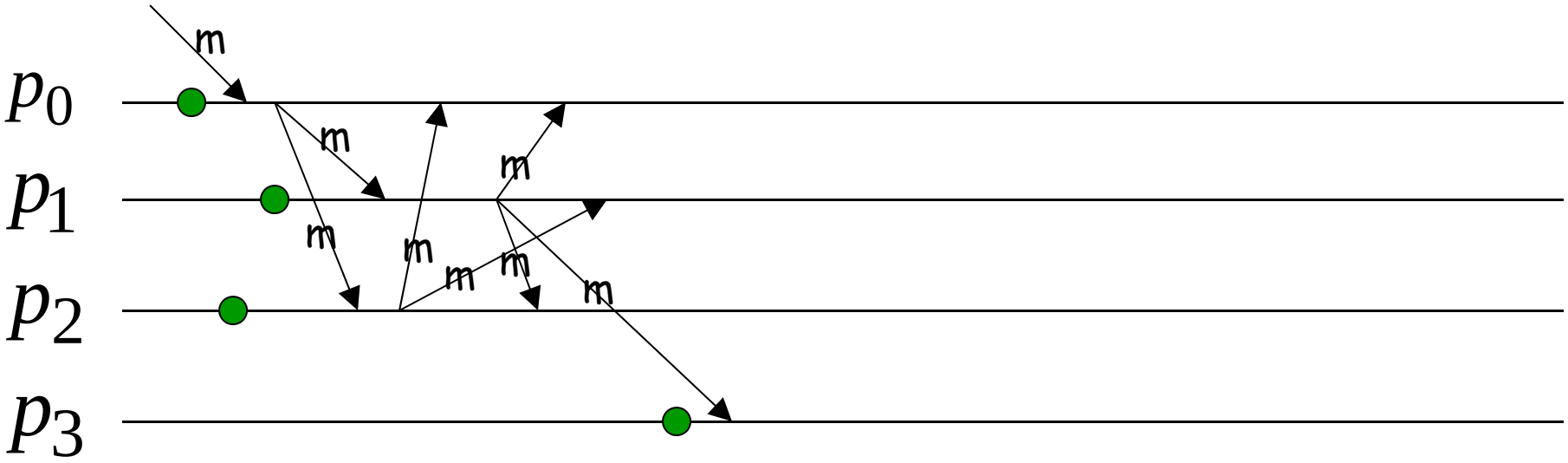
$$S = \{p_0\}$$



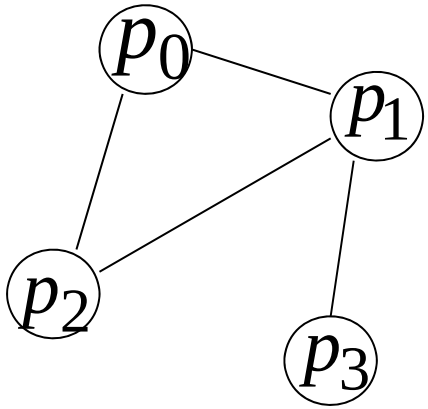


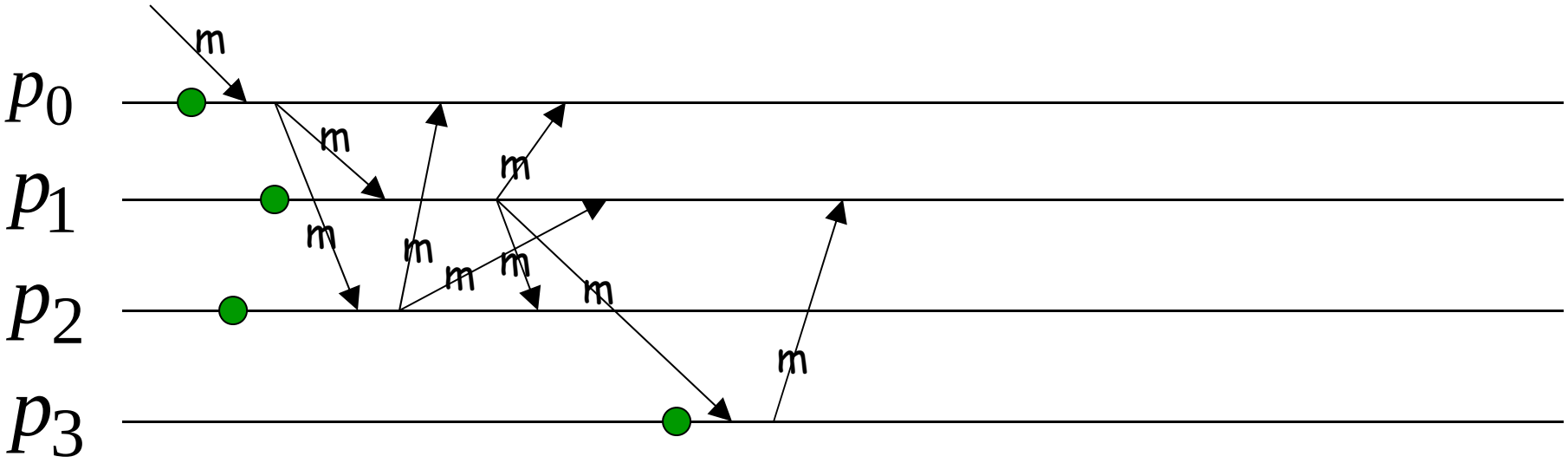
$$S = \{p_0\}$$



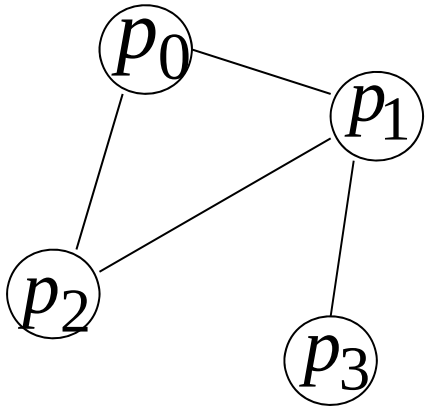


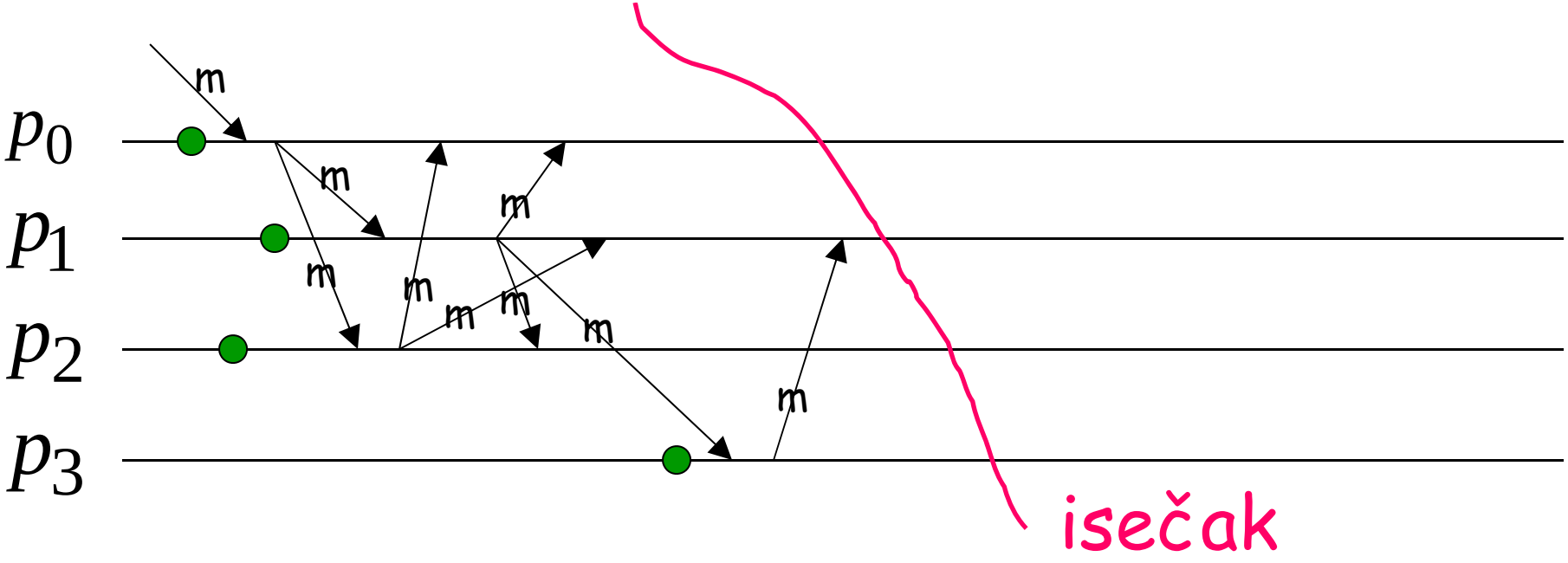
$$S = \{p_0\}$$



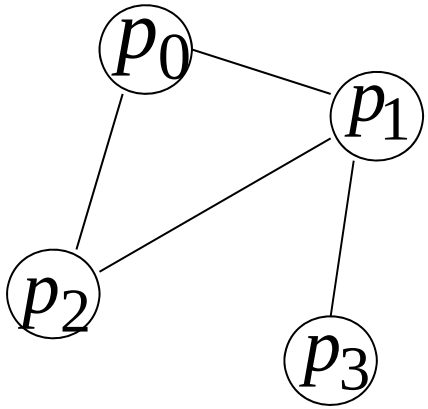


$$S = \{p_0\}$$





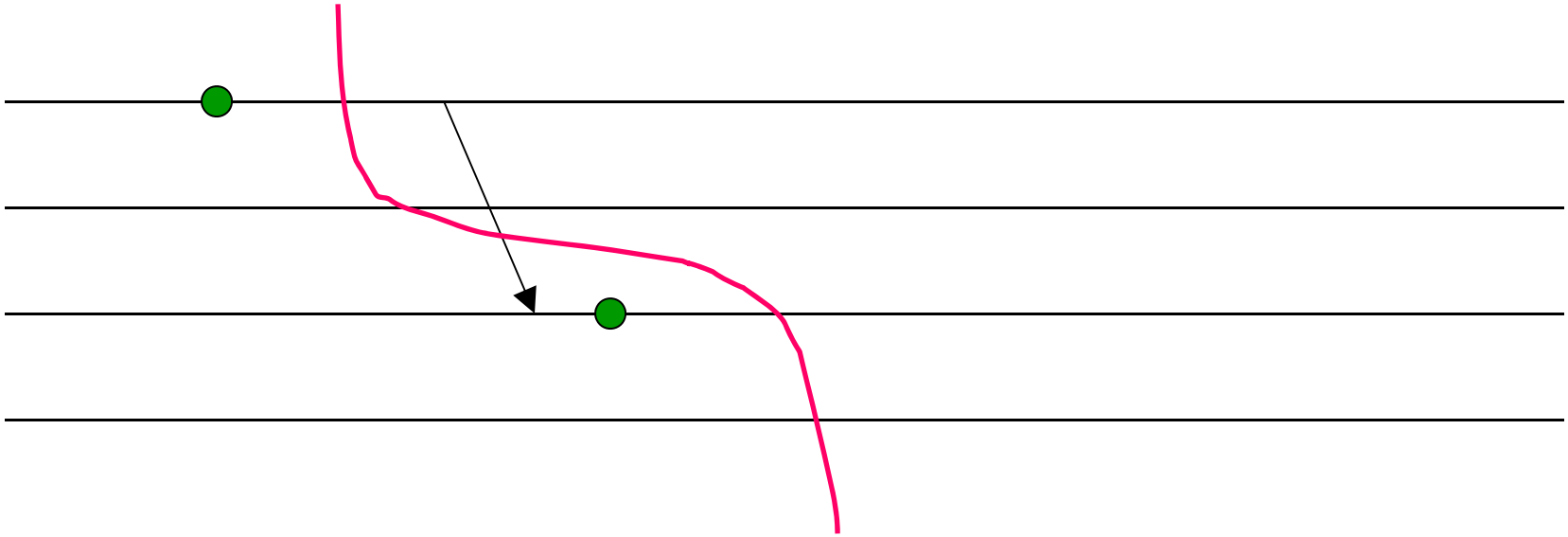
$$S = \{p_0\}$$



Teorema: Isečak dobijen pomoću ovog algoritma je konzistentan

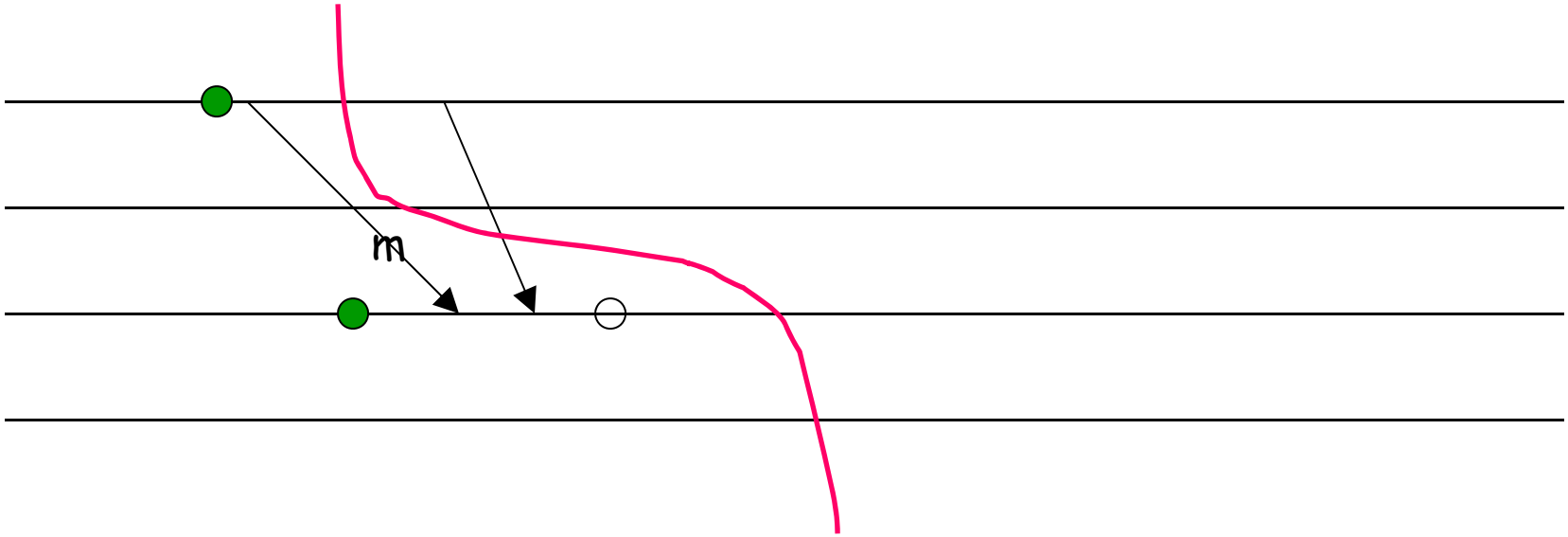
Dokaz: Kontradikcijom

Pred. da je isečak nekonzistentan



(predpostavljamo FIFO)

Moralo bi da bude:

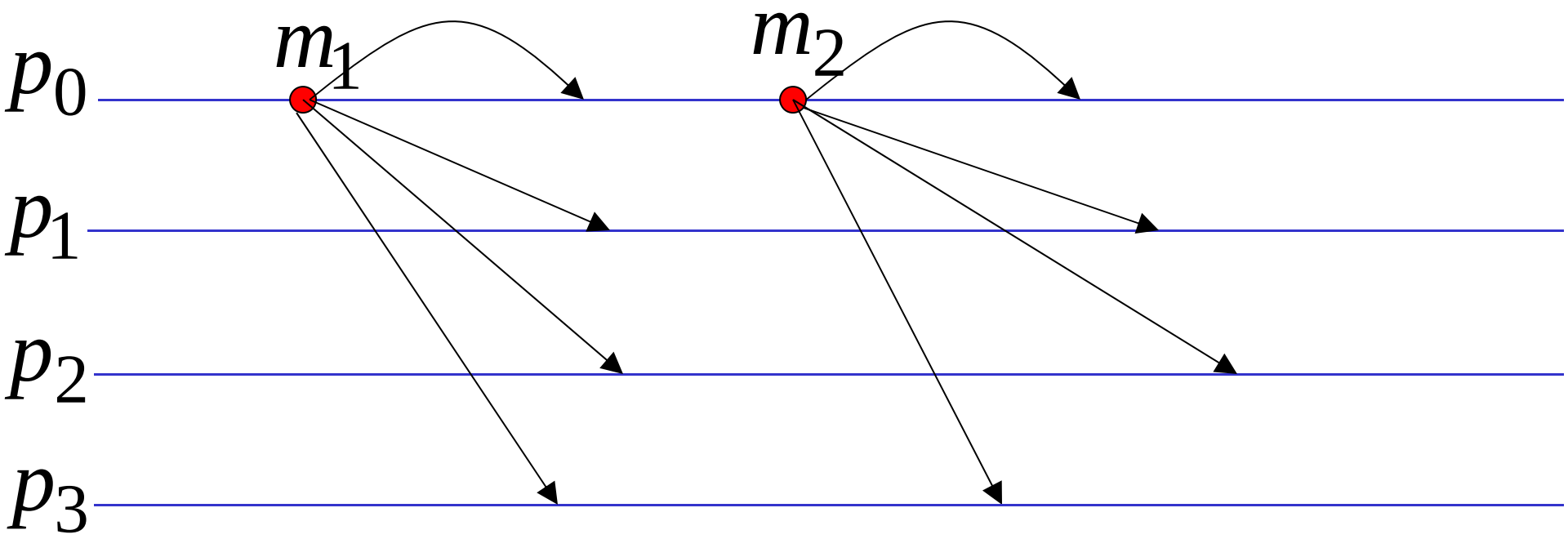


Nemoguće!

Kraj dokaza

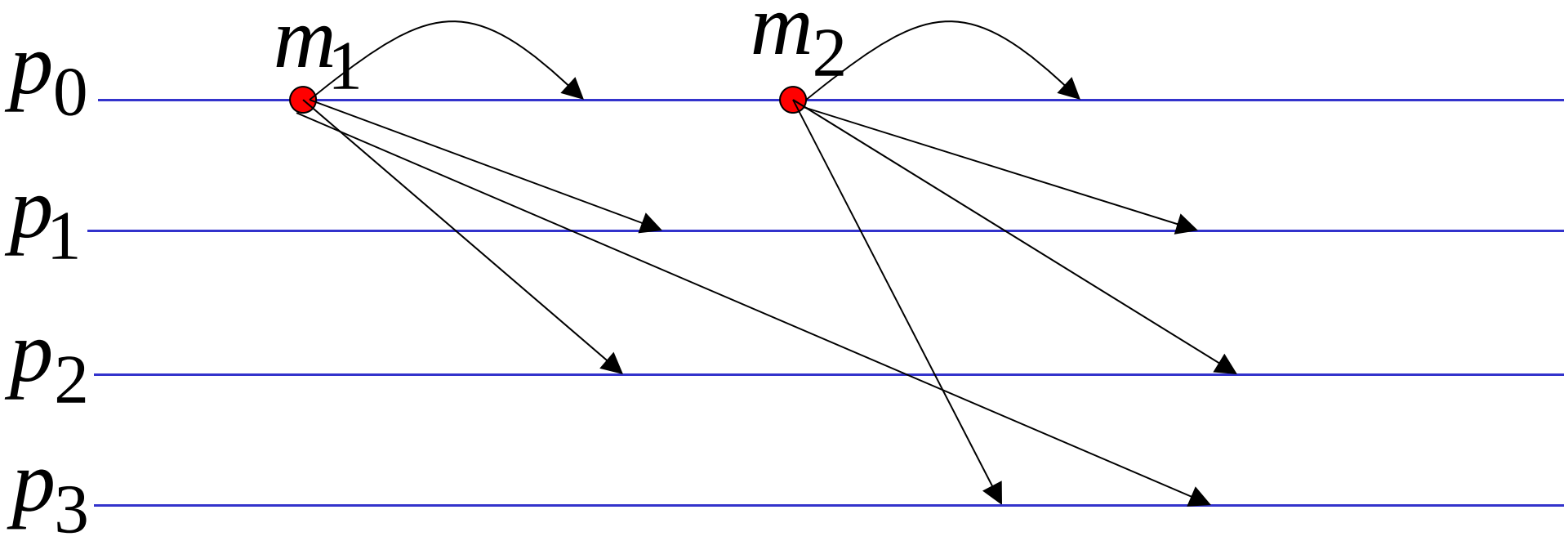
FIFO slanje svima (FIFO-Broadcast)

FIFO redosled

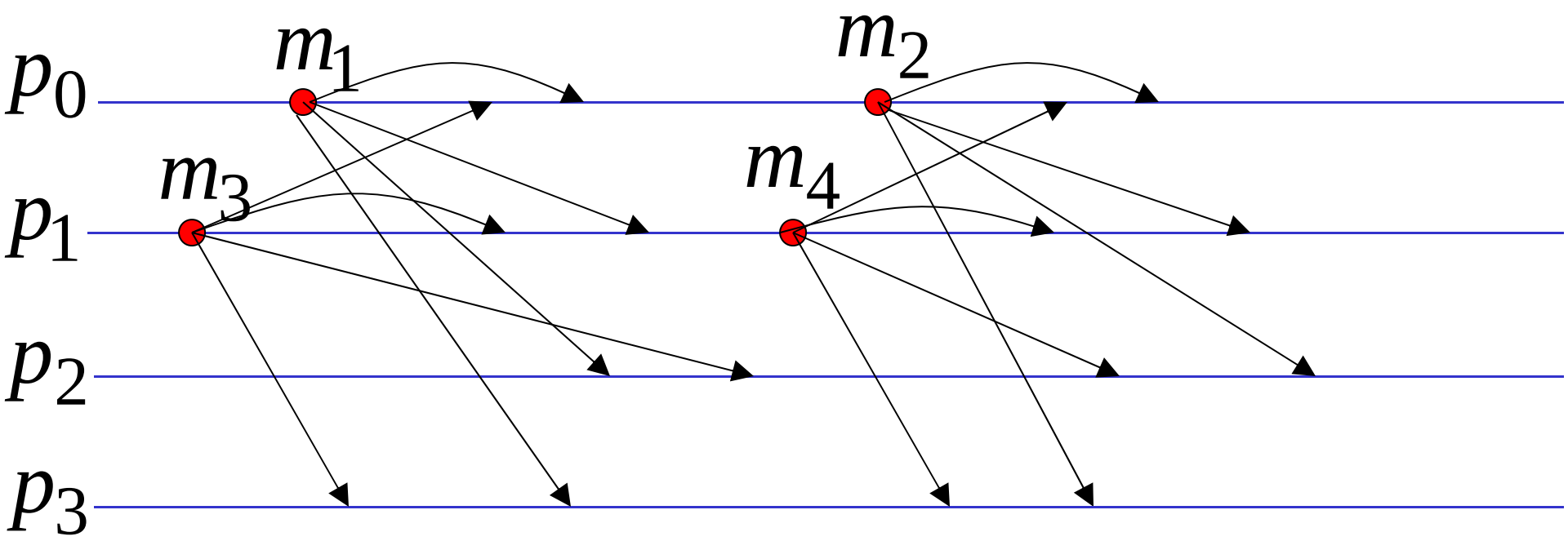


Poruke od p_i se primaju u redosledu u kom su slane od p_i

Nije FIFO-redosled

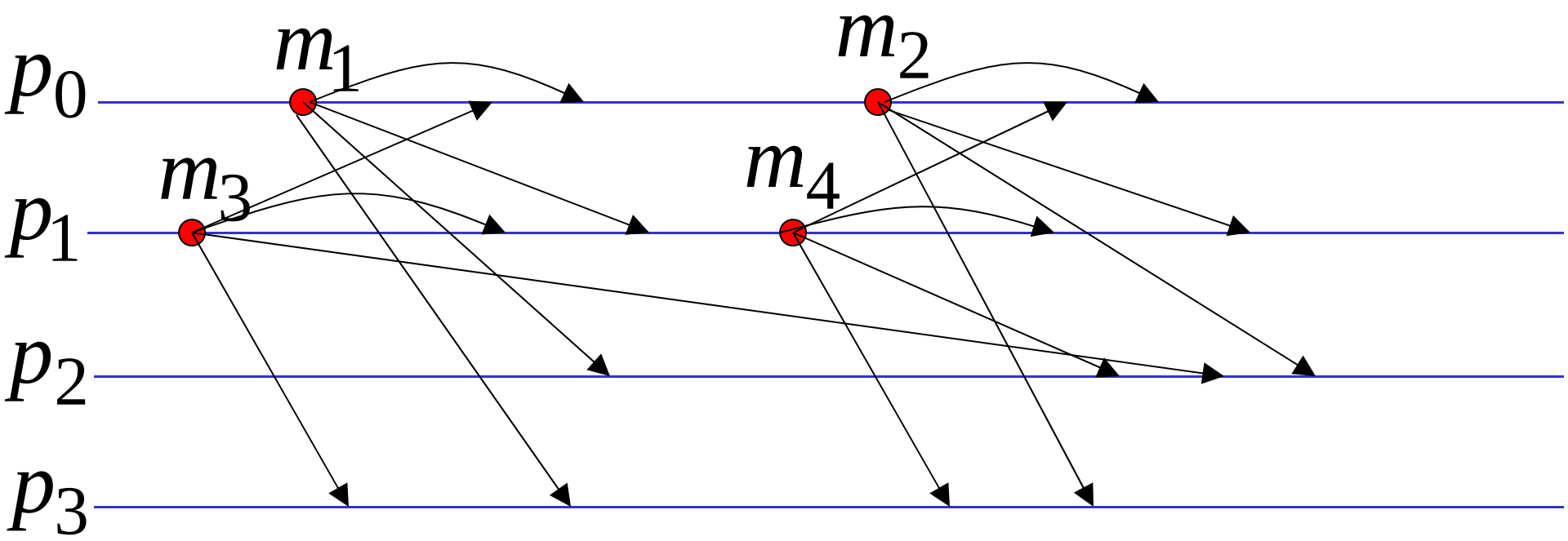


FIFO redosled

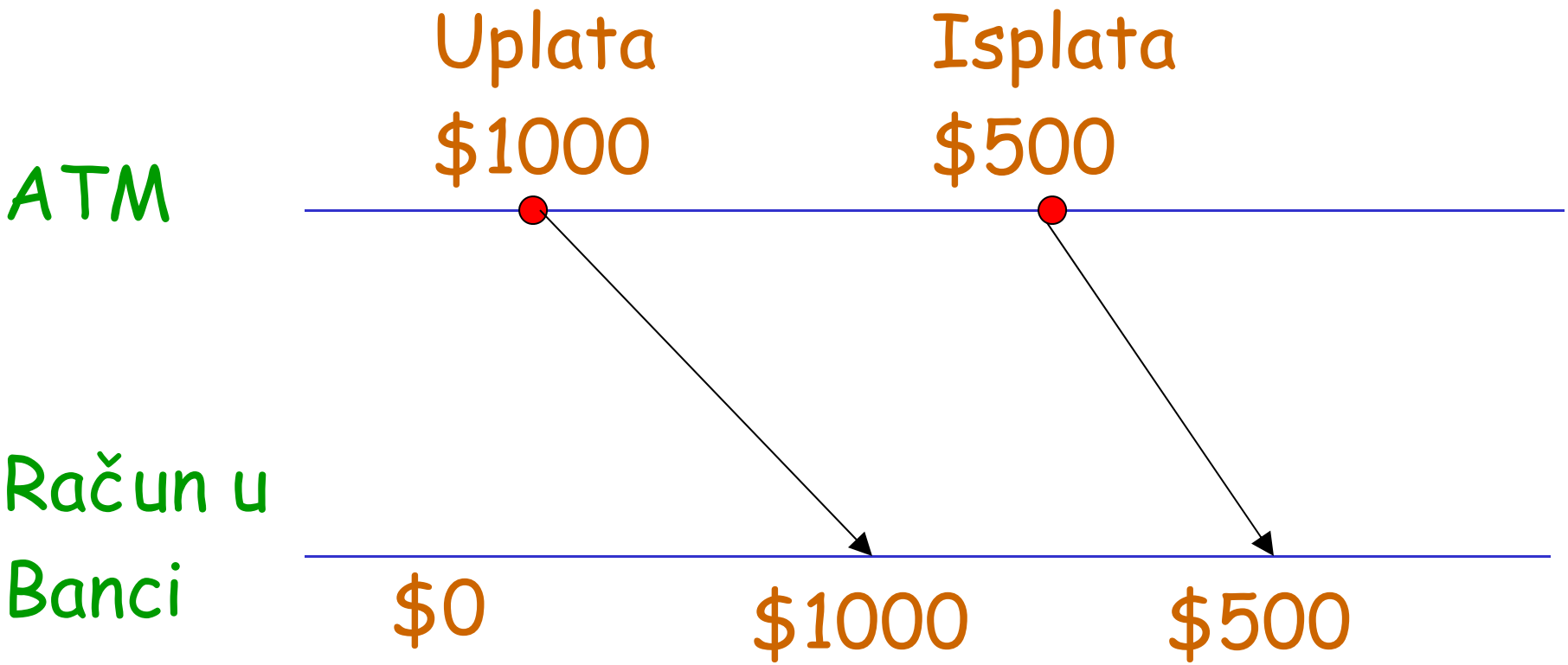


Poruke od različitih procesora mogu biti primljene u različitom redosledu

Nije FIFO-redosled

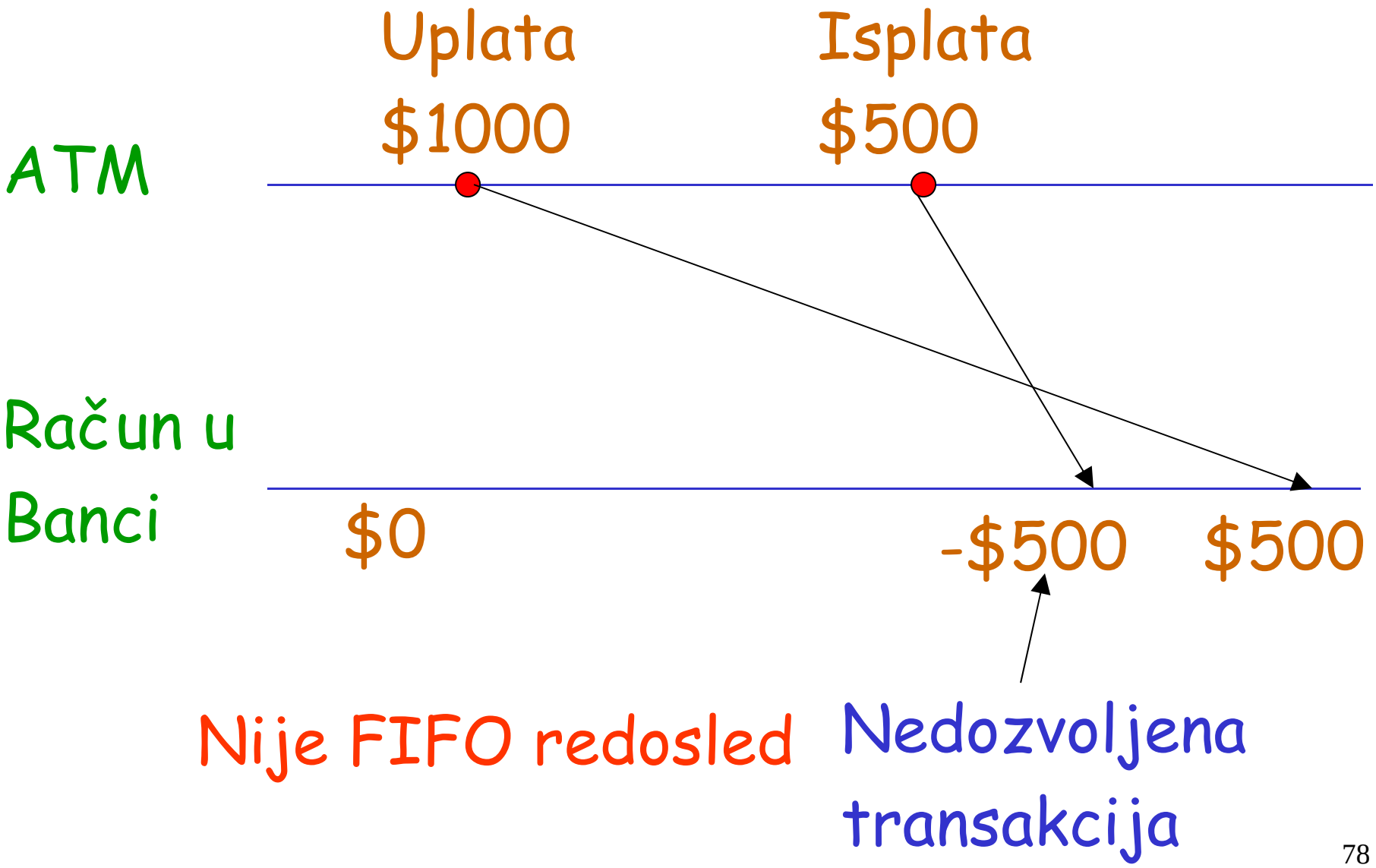


Zašto je FIFO važan?

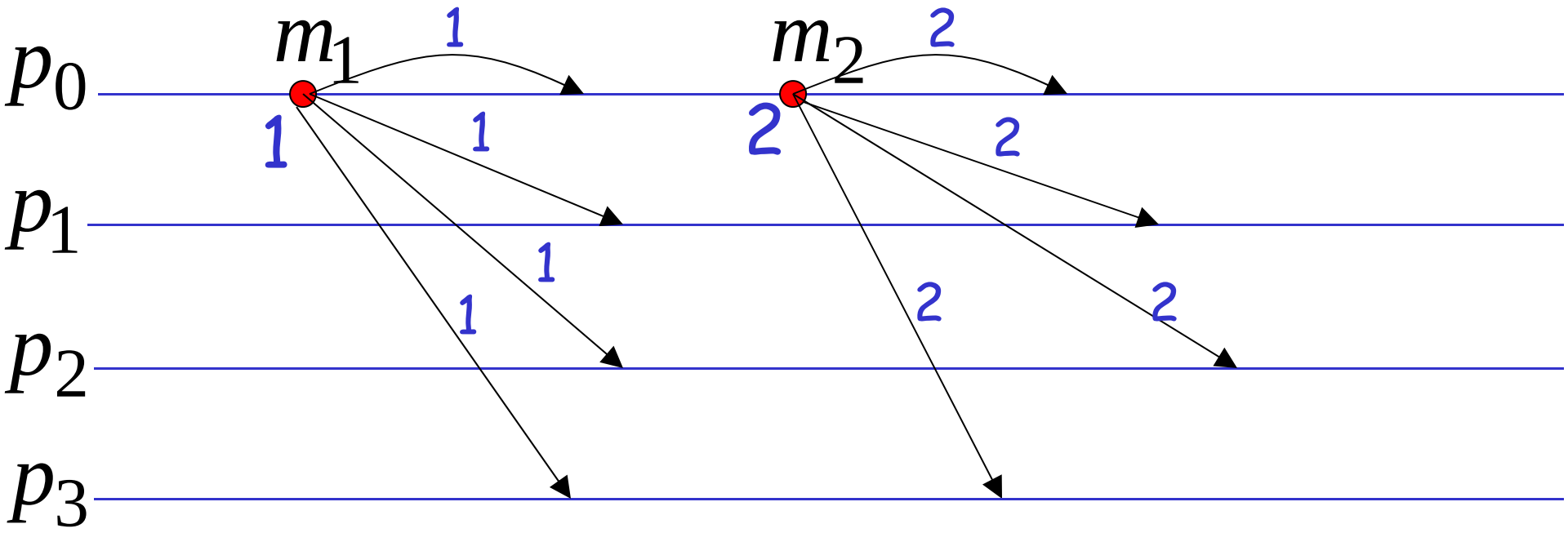


FIFO redosled

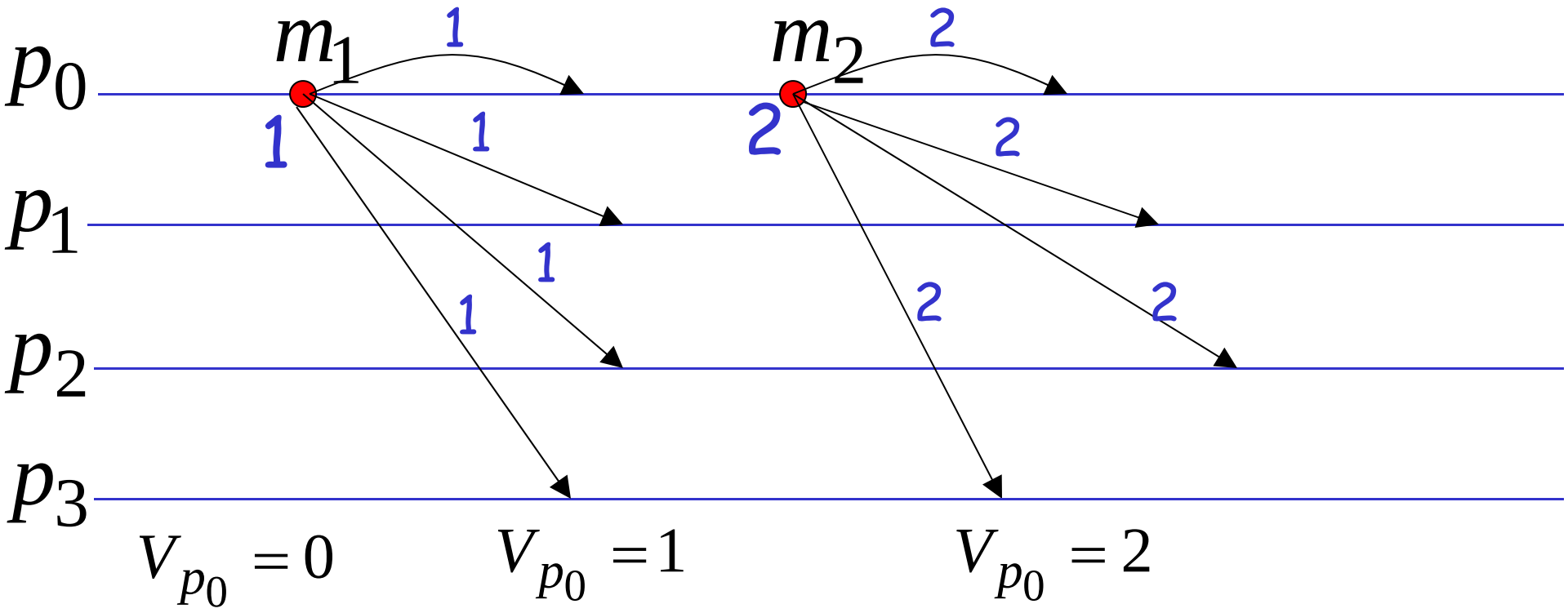
Zašto je FIFO važan?



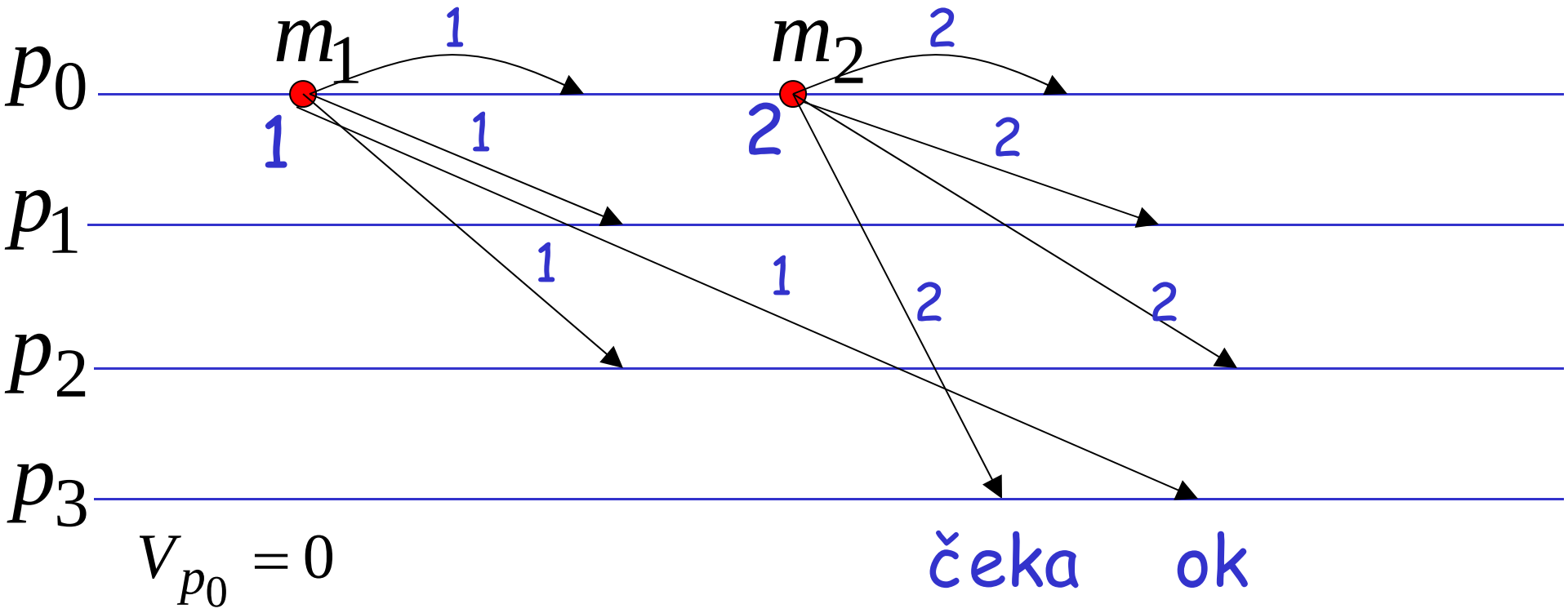
Jednostavan FIFO algoritam



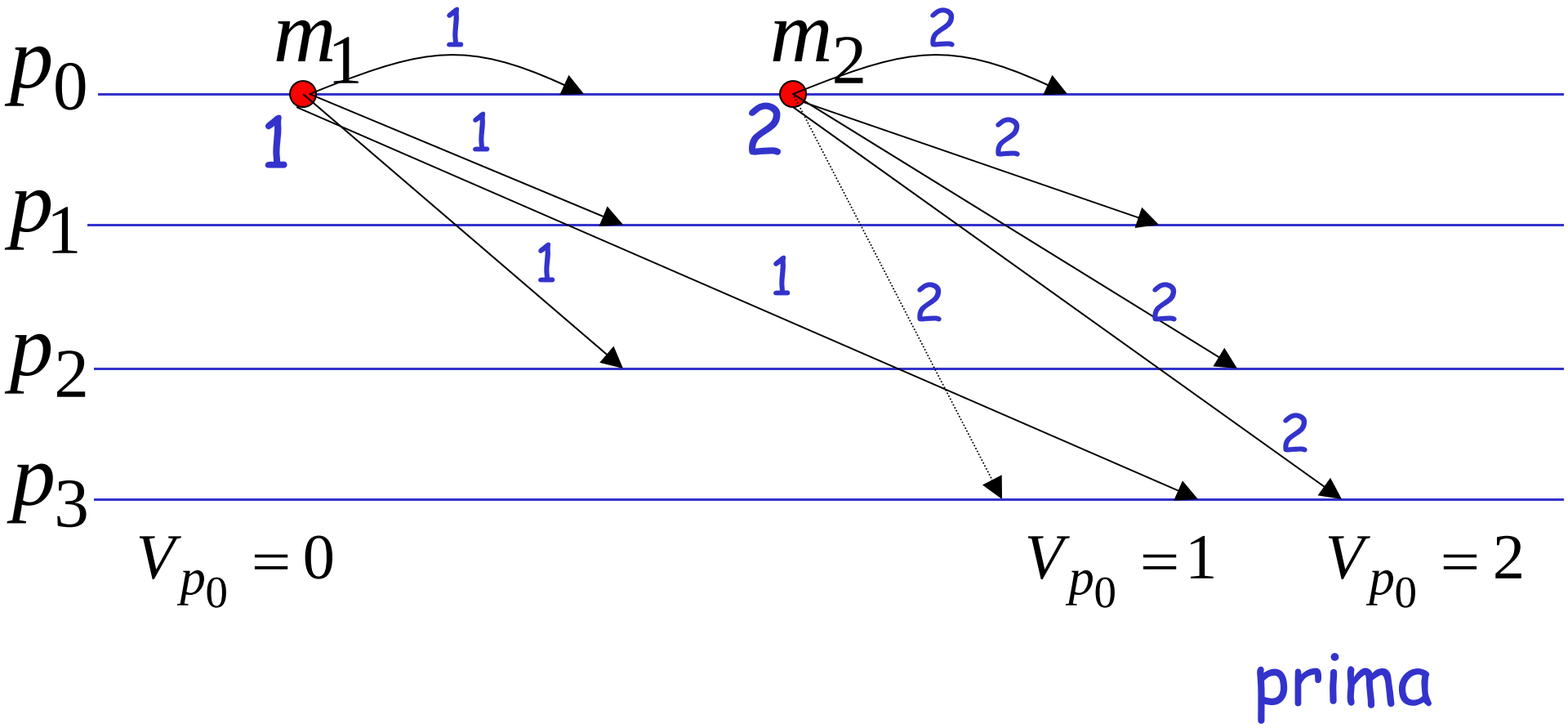
Procesor inkrementira brojač
za svako slanje svima



Procesor održava zapis najveće vrednosti primljene od drugih procesora

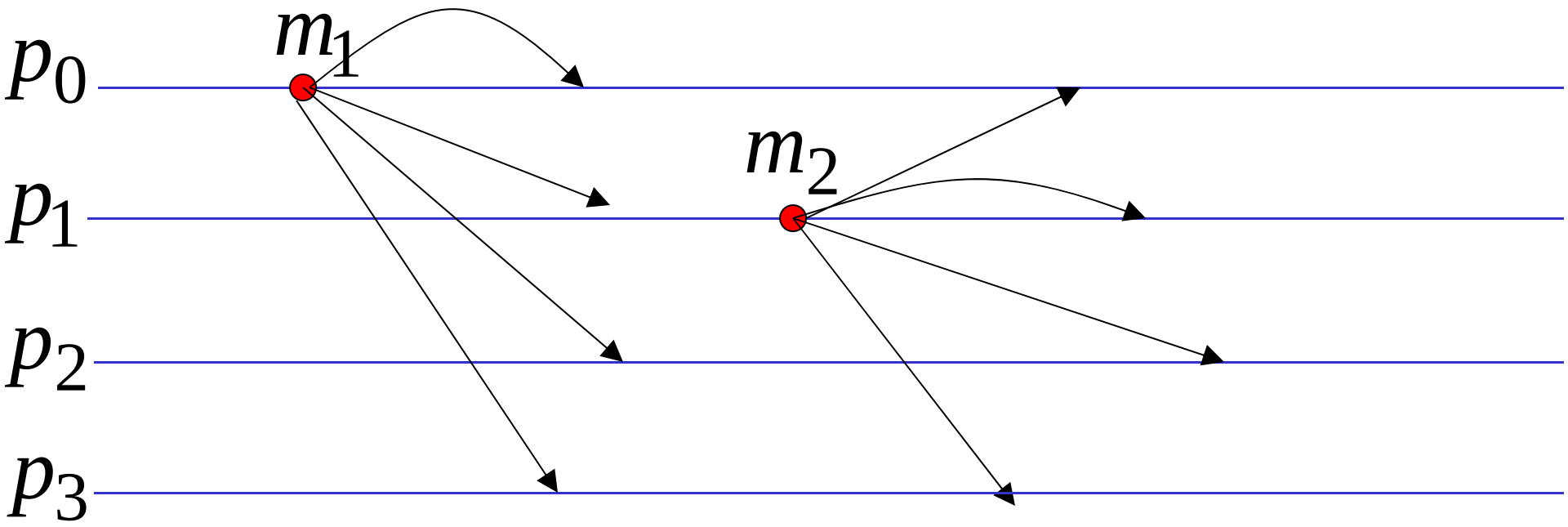


Čeka dok ne budu primljene sve poruke sa manjim vrednostima od istog pošiljaoca


p_0
p_1
p_2
p_3
m_1
m_2
1
2
$V_{p_0} = 0$
$V_{p_0} = 1$
$V_{p_0} = 2$
prima

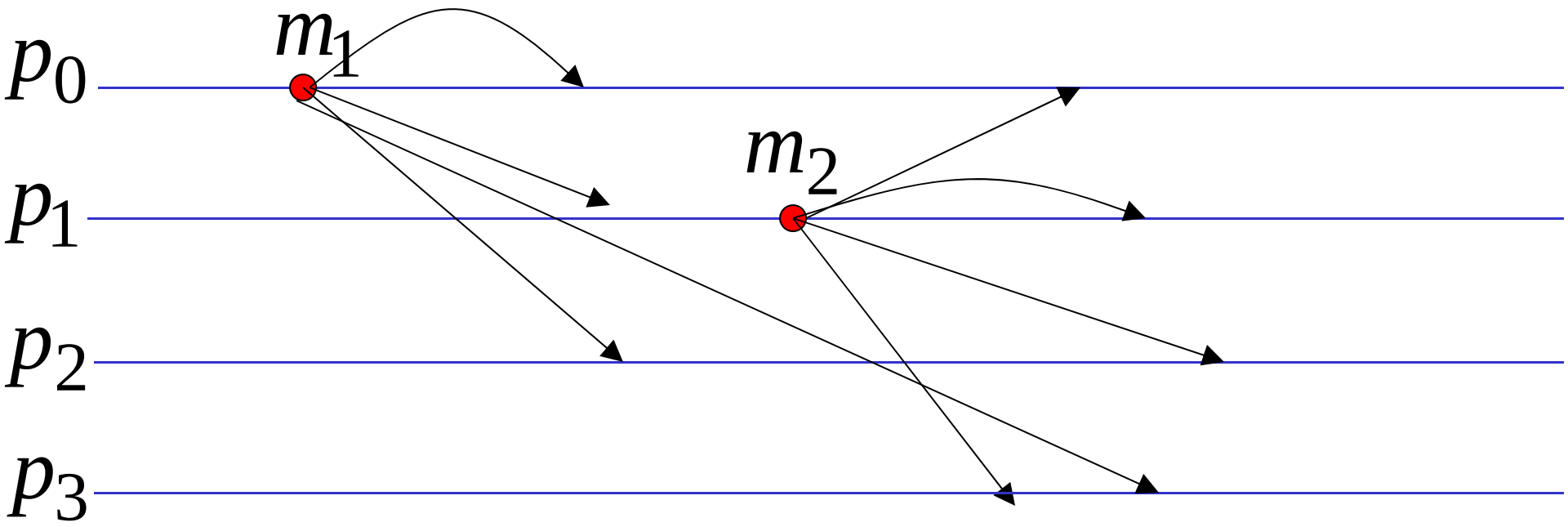
Totalni redosled – Slanje svima

Totalni redosled (Total Order)

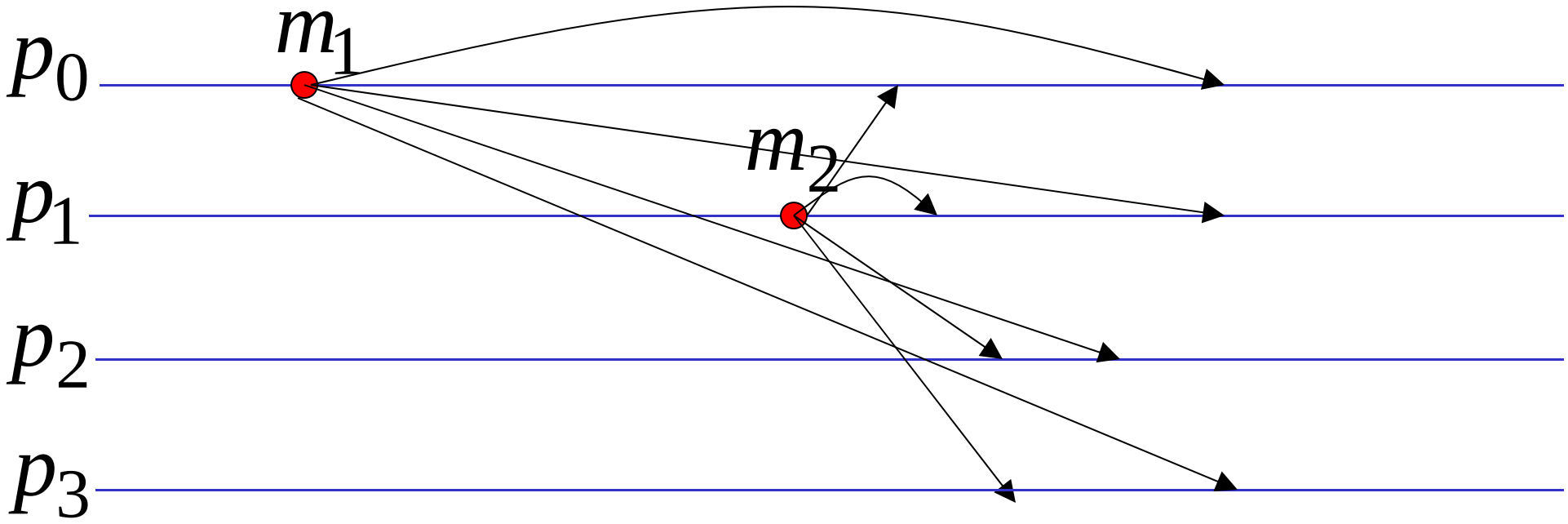


Poruke se primaju u
istom redosledu u svakom procesoru

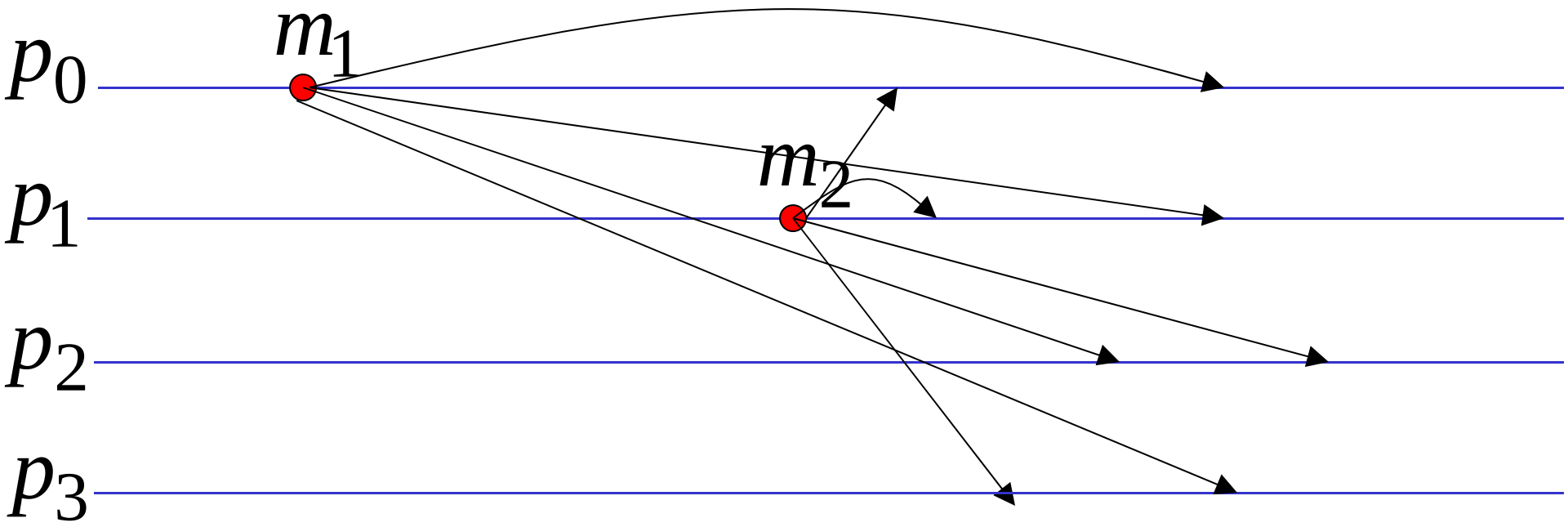
Nije totalni redosled



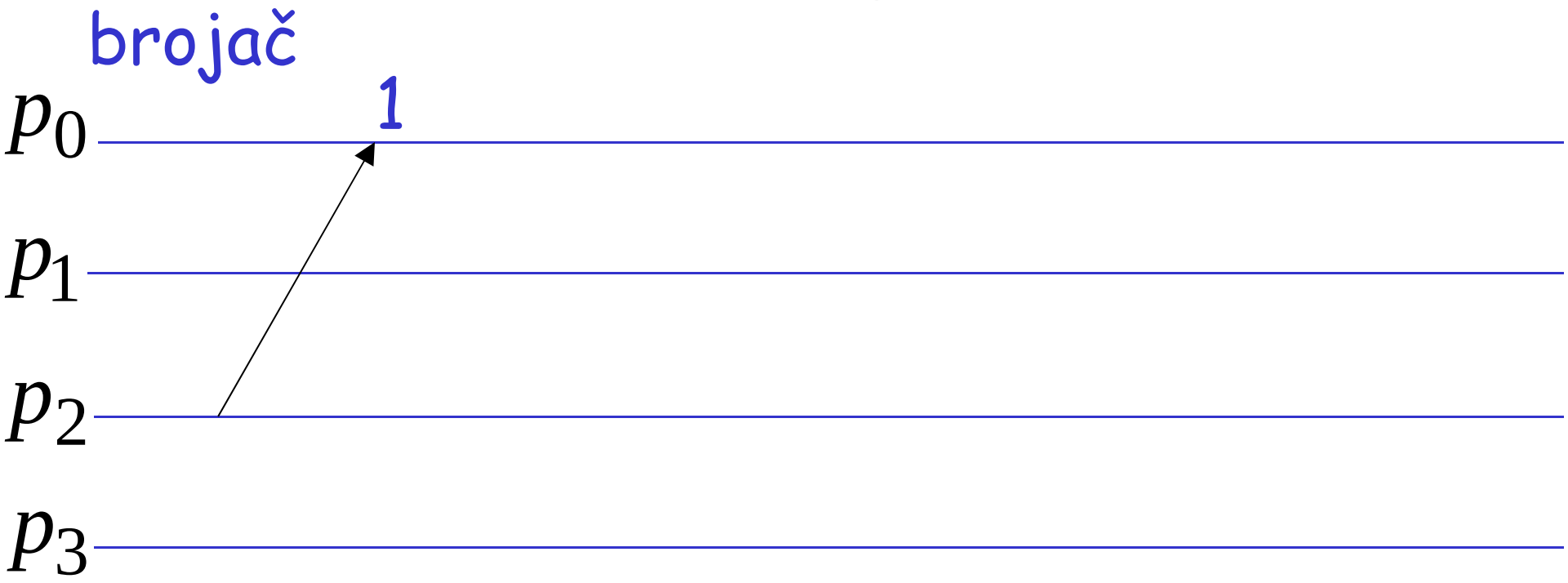
Totalni redosled



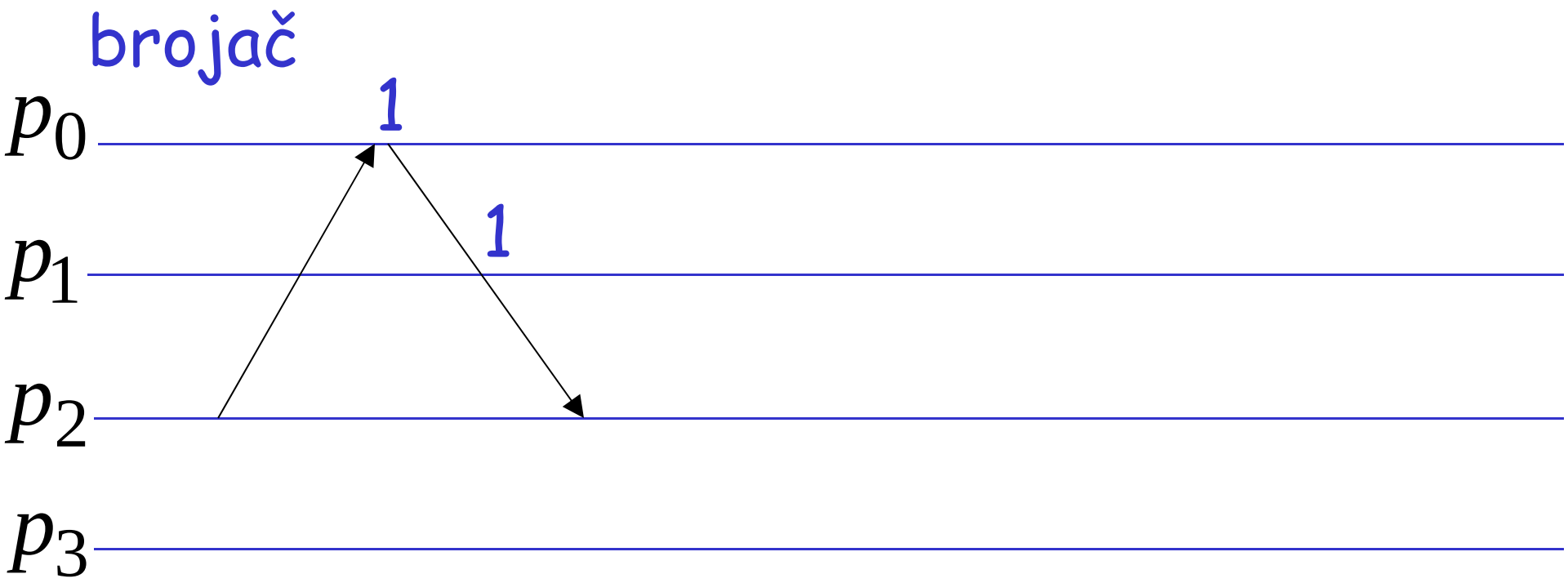
Nije totalni redosled



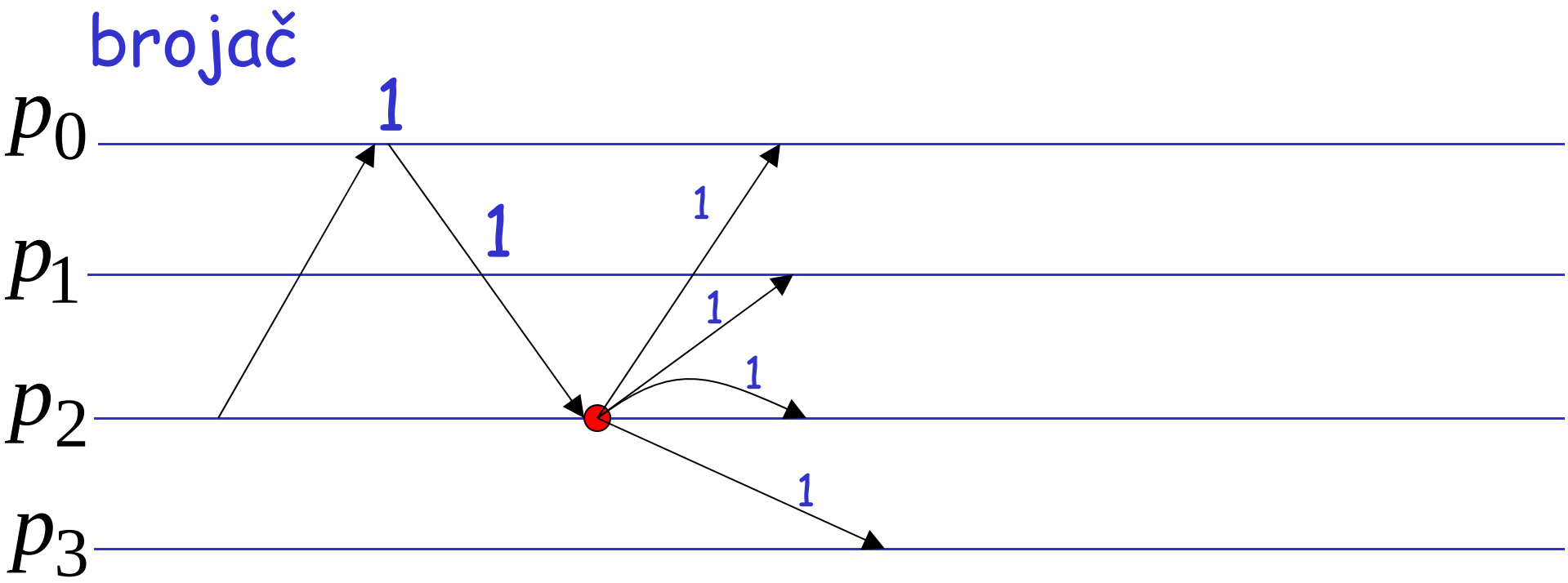
Asimetričan algoritam



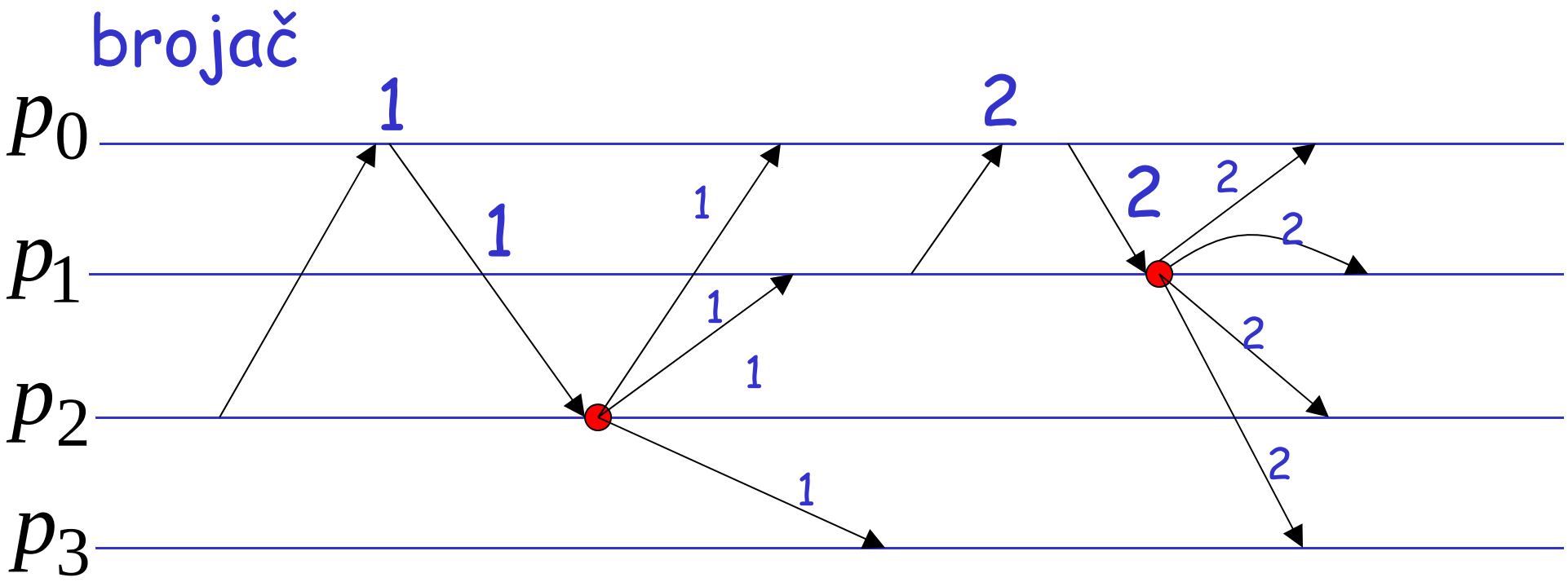
Zahteva vrednost brojača

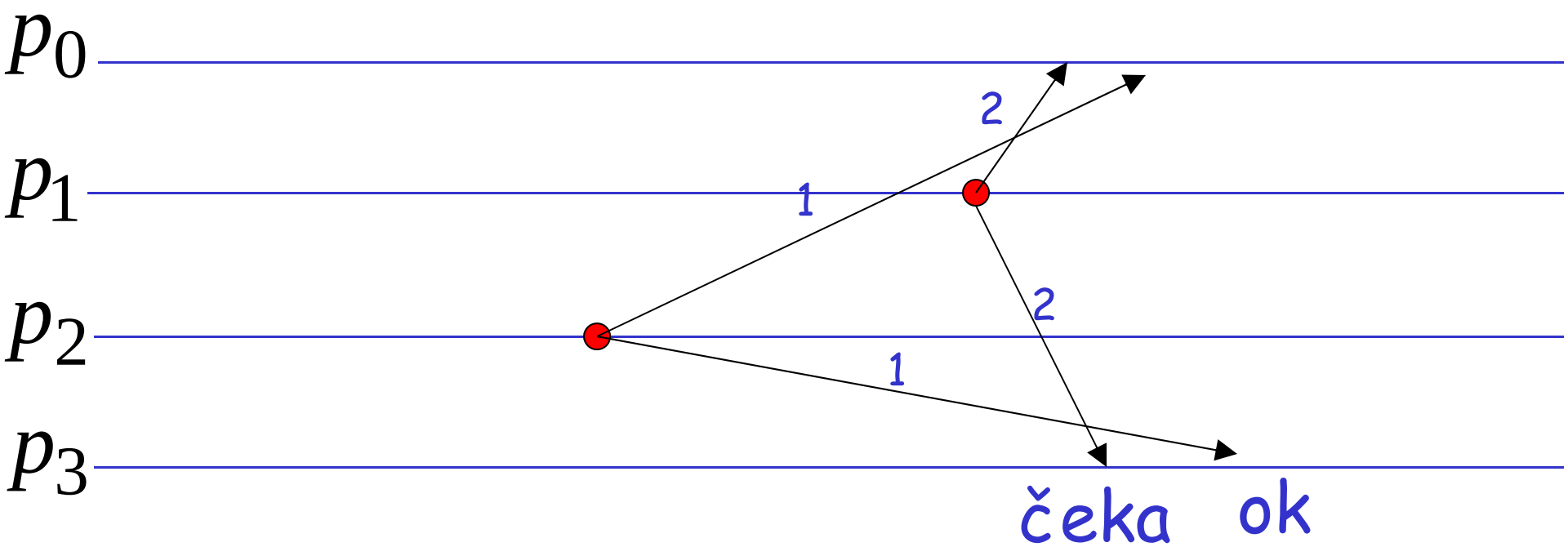


Prima vrednost brojača

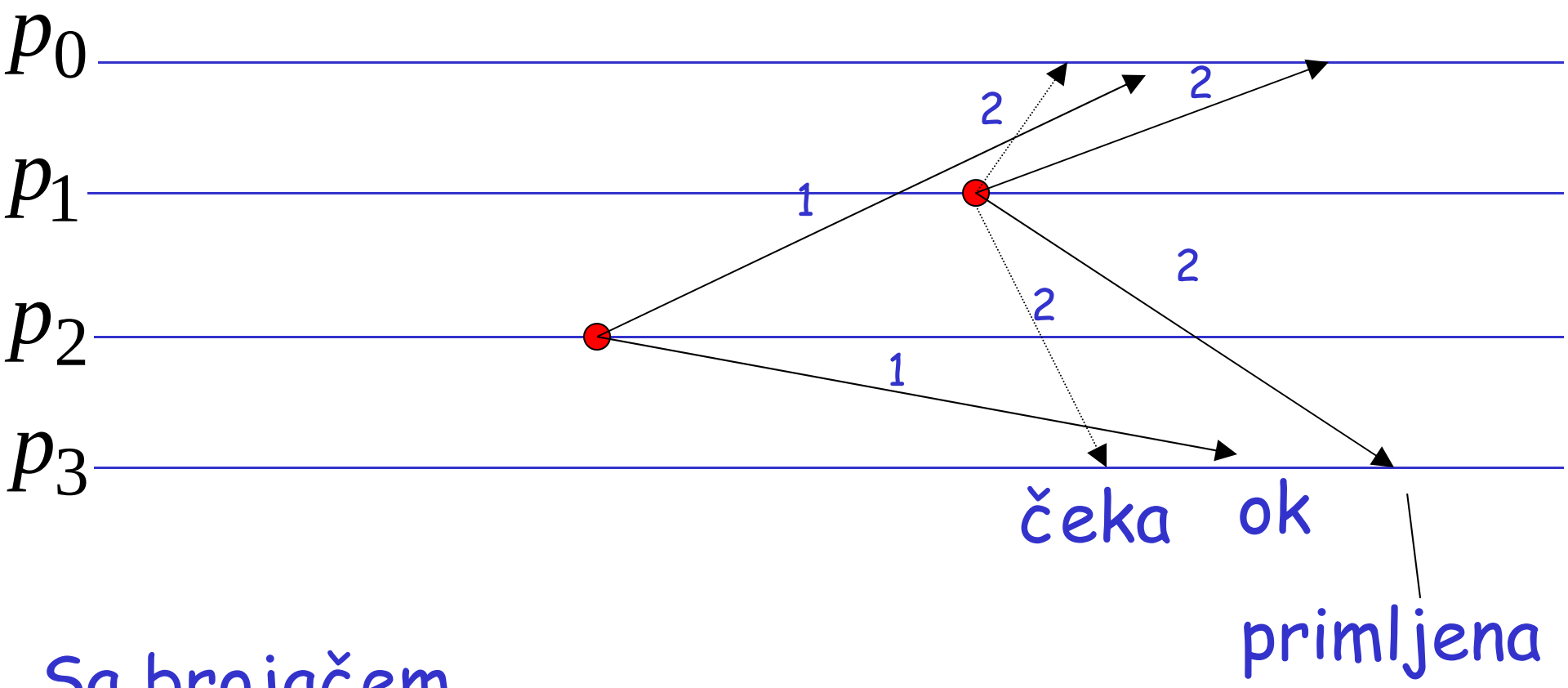


Šalje svima


brojač
p_0
p_1
p_2
p_3
1
1
1
1
1
2
2
2
2
2
2



Sa brojačem,
poruke upućene svima su u totalnom redosledu



Sa brojačem,

poruke upućene svima su u totalnom redosledu

Simetričan algoritam

Osnova mu je FIFO

Koristi koncept vektorskih satova

Svaki proces ima brojač

Prilikom slanja svima (broadcast):

proces šalje vrednost brojača svakom učesniku

Po prijemu poruke:

Ako je primljena vrednost brojača veća od lokalne vrednosti, onda:

- Ažuriraj lokalni brojač na novu vred.
- Pošalji svima drugima novu vrednost brojača

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

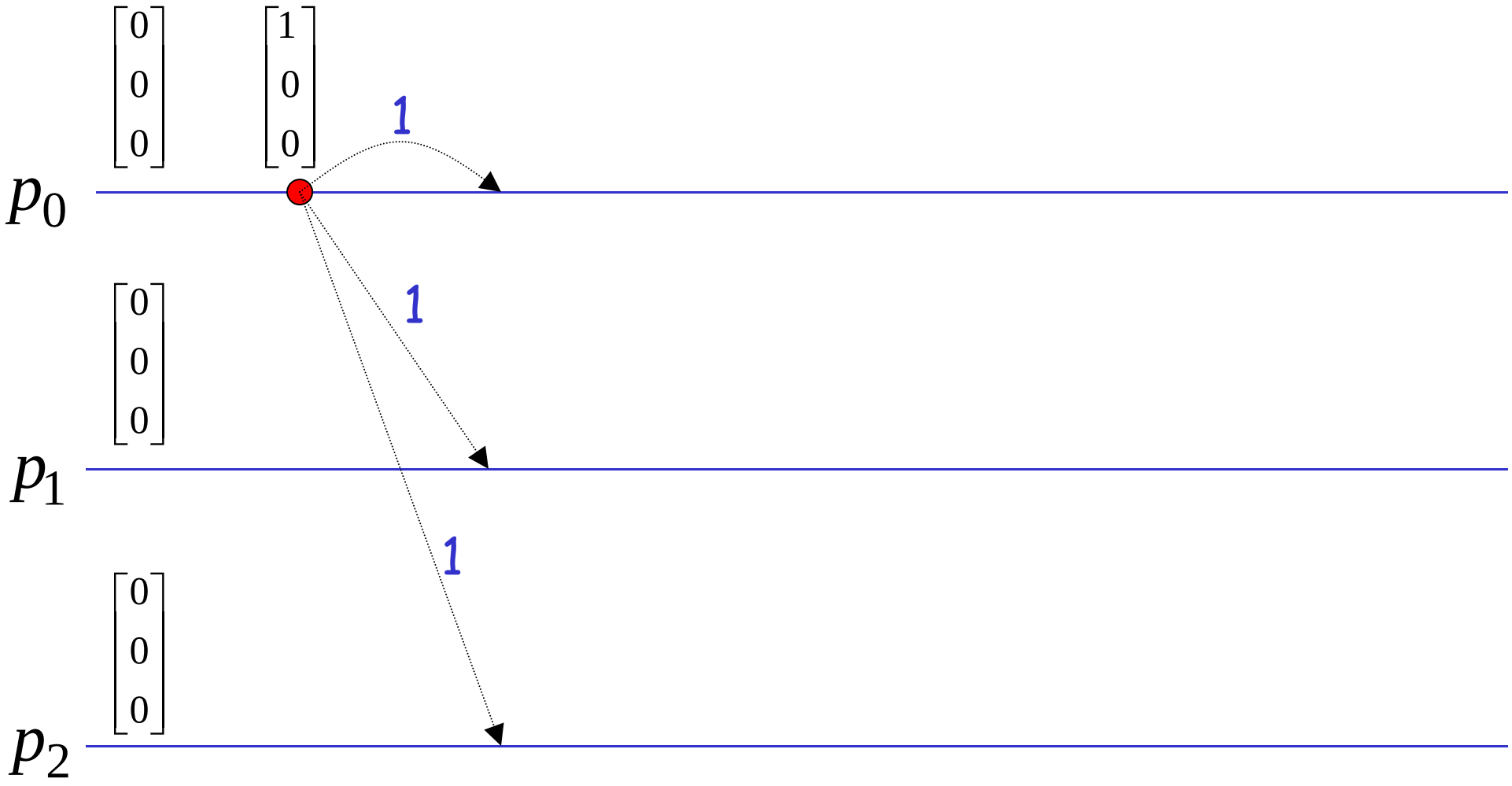
p_0

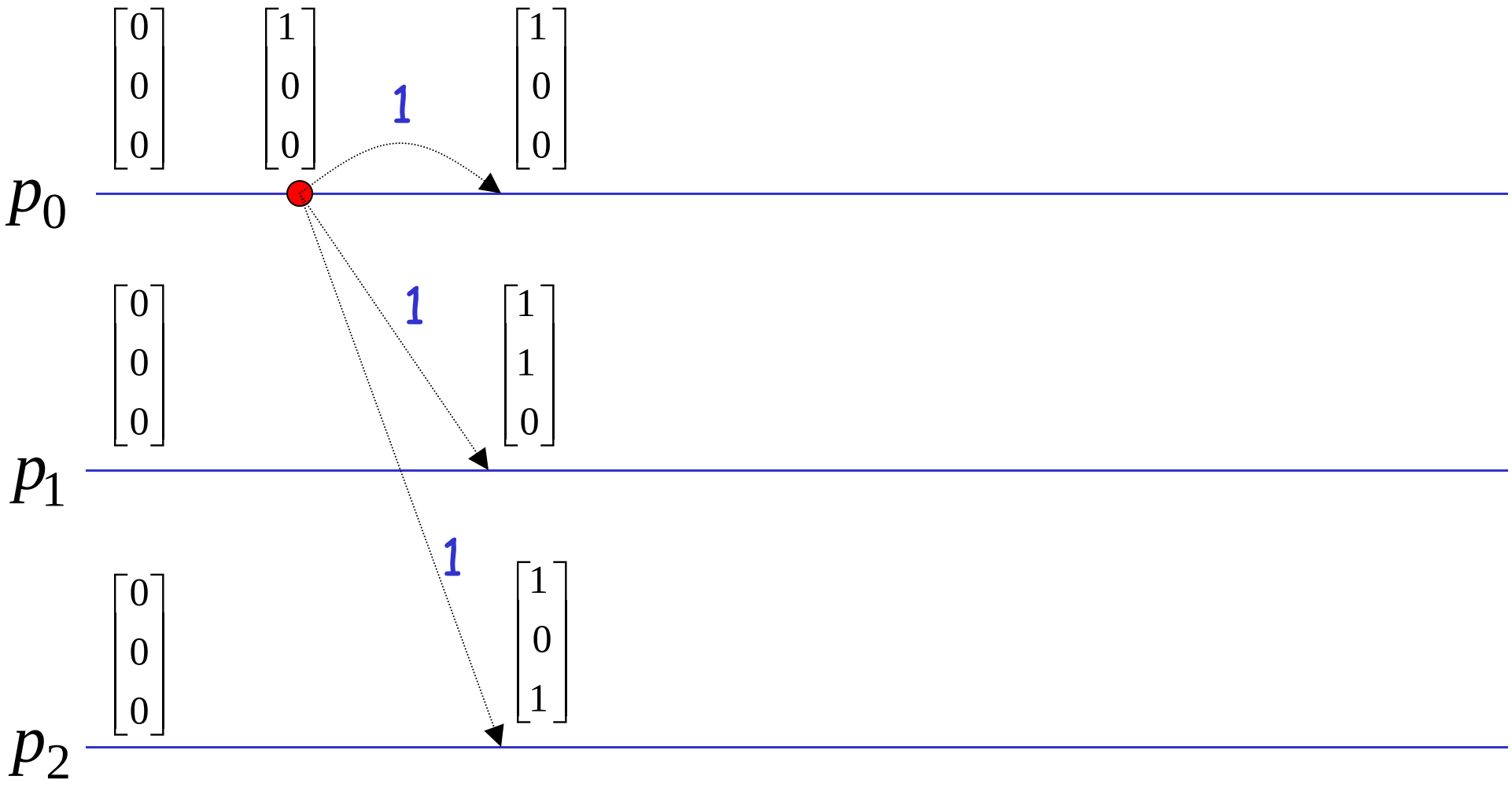
$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

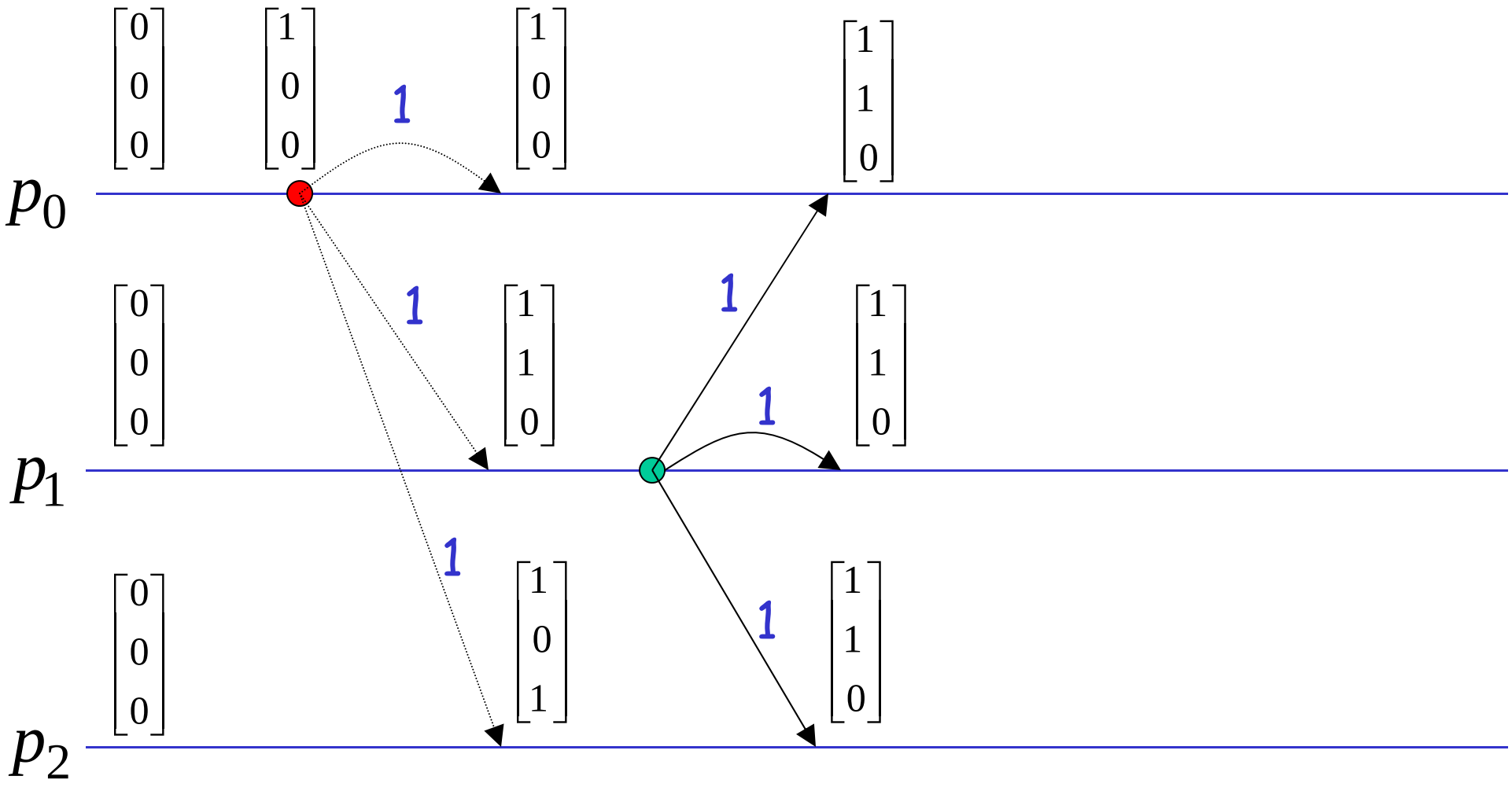
p_1

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

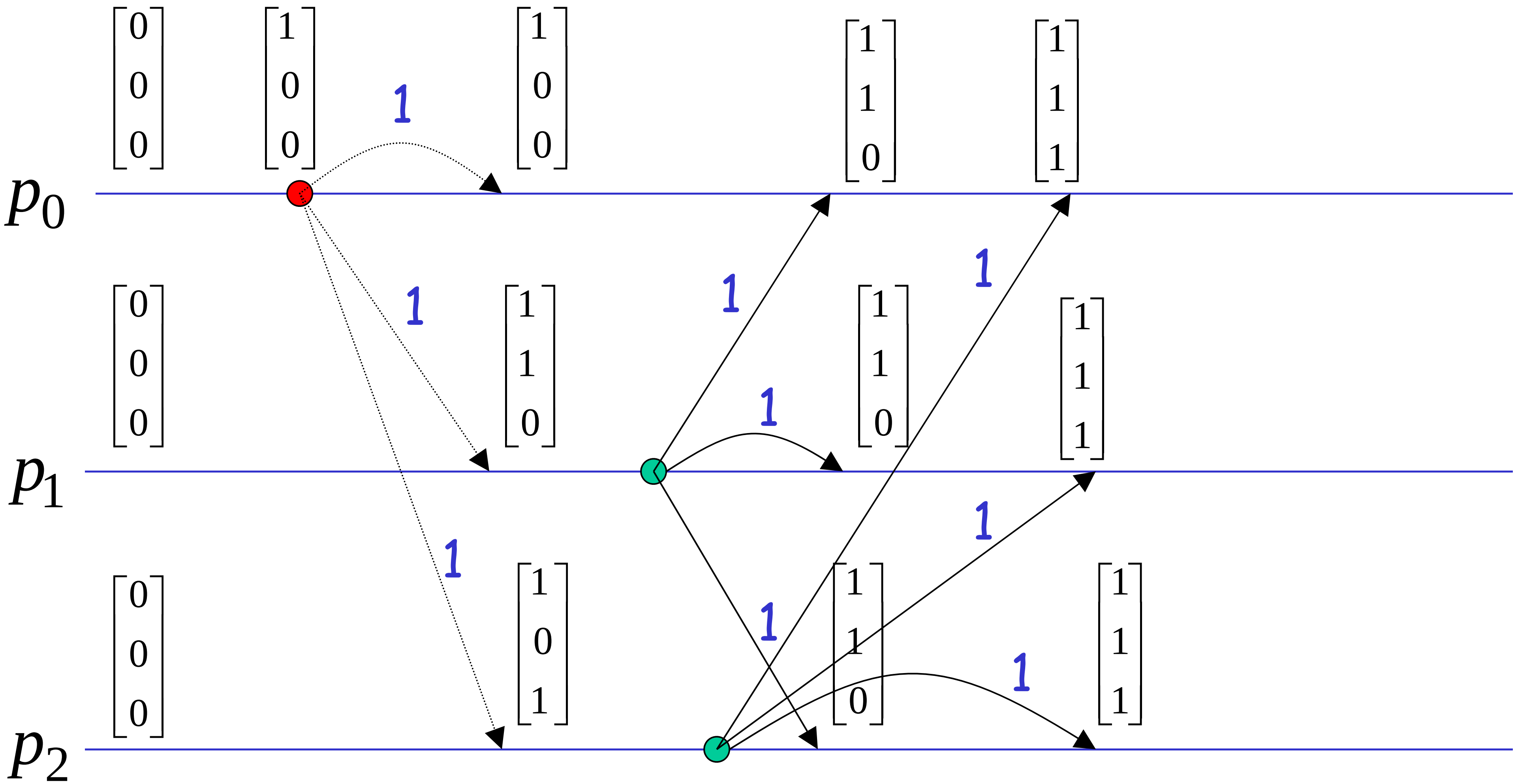
p_2

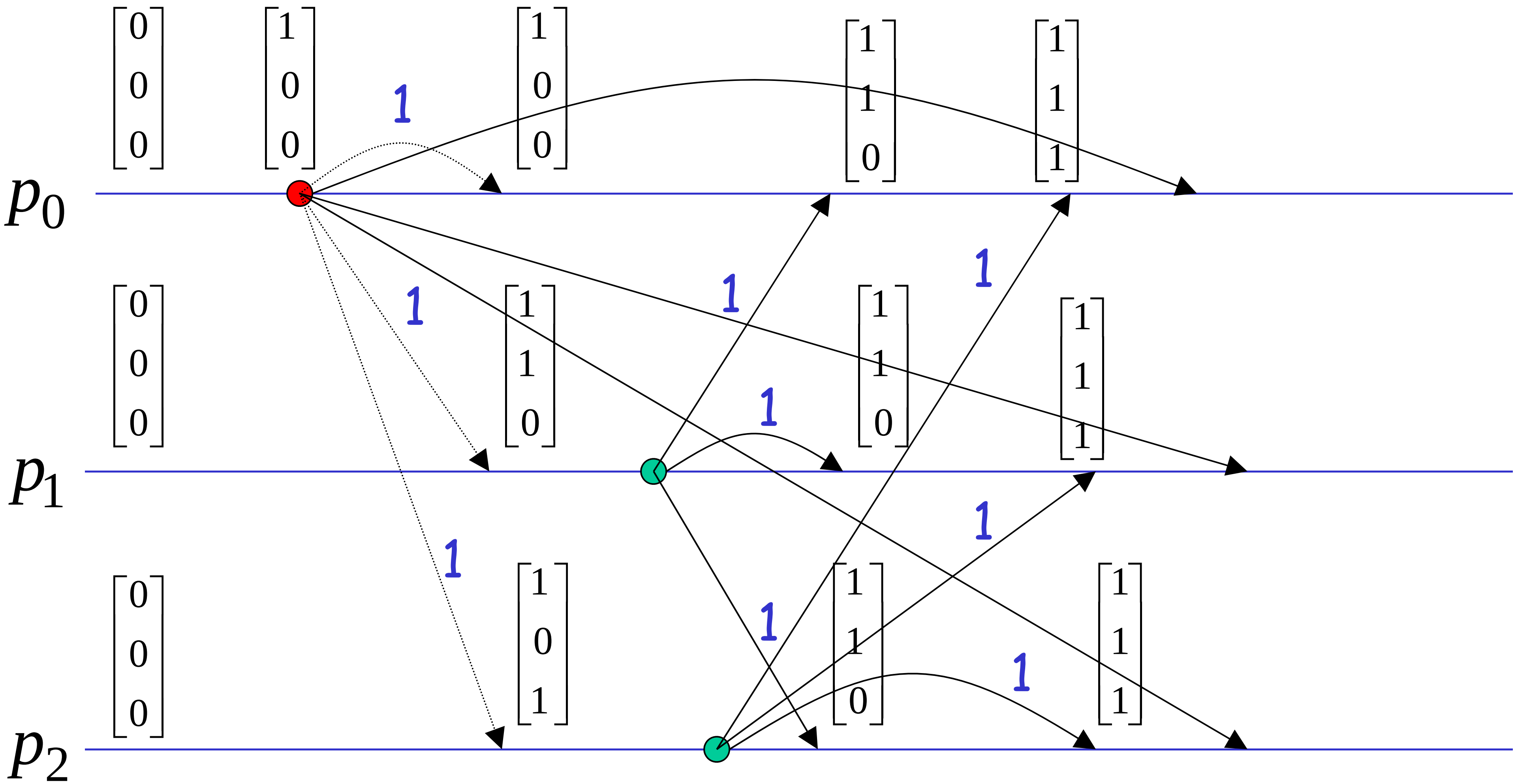

$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
1
p_0
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
1
p_1
1
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
p_2


$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
1
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
p_0
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
1
$\begin{bmatrix}1\\1\\0\end{bmatrix}$
p_1
1
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
p_2

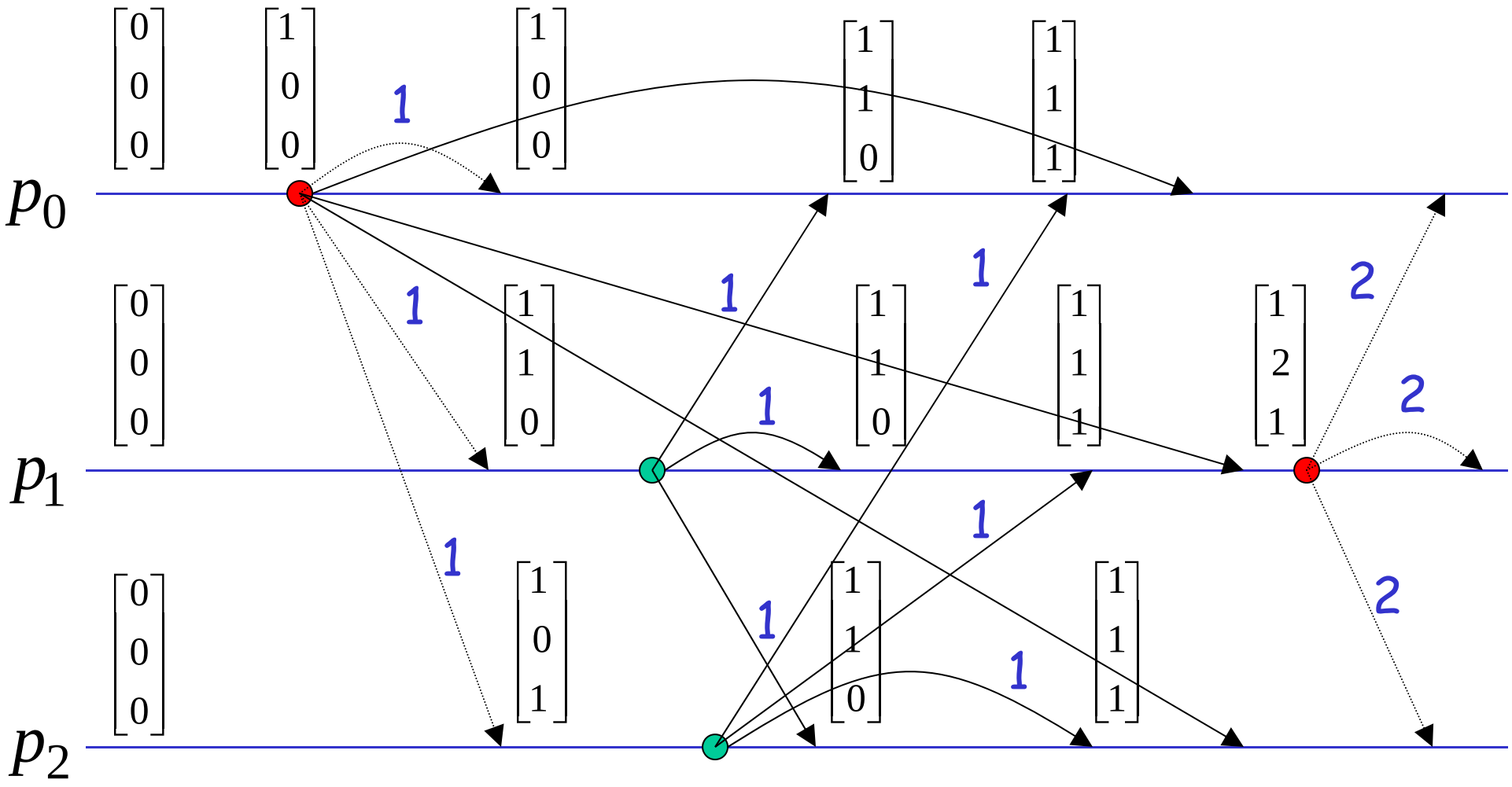


Specijalna poruka upućena svima


p_0
p_1
p_2
$\begin{bmatrix}0\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\0\\1\end{bmatrix}$ $\begin{bmatrix}1\\1\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$
1
1
1
1
1
1
1
1
1



Stvarni prijem normalne poruke upućene svima



Sledeće normalno slanje svima

Lokalni vektor od: p_i

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \end{bmatrix}$$

p_i čuva normalne poruke upućene svima u red sortiran po broju poruke. Da bi primio poruku br. T, mora predhodno primiti sve predhodne poruke, a tada je a_j bar jednako T, za sve j, i tada se iz reda vade sve poruke sa brojem manjim ili jednakim T.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

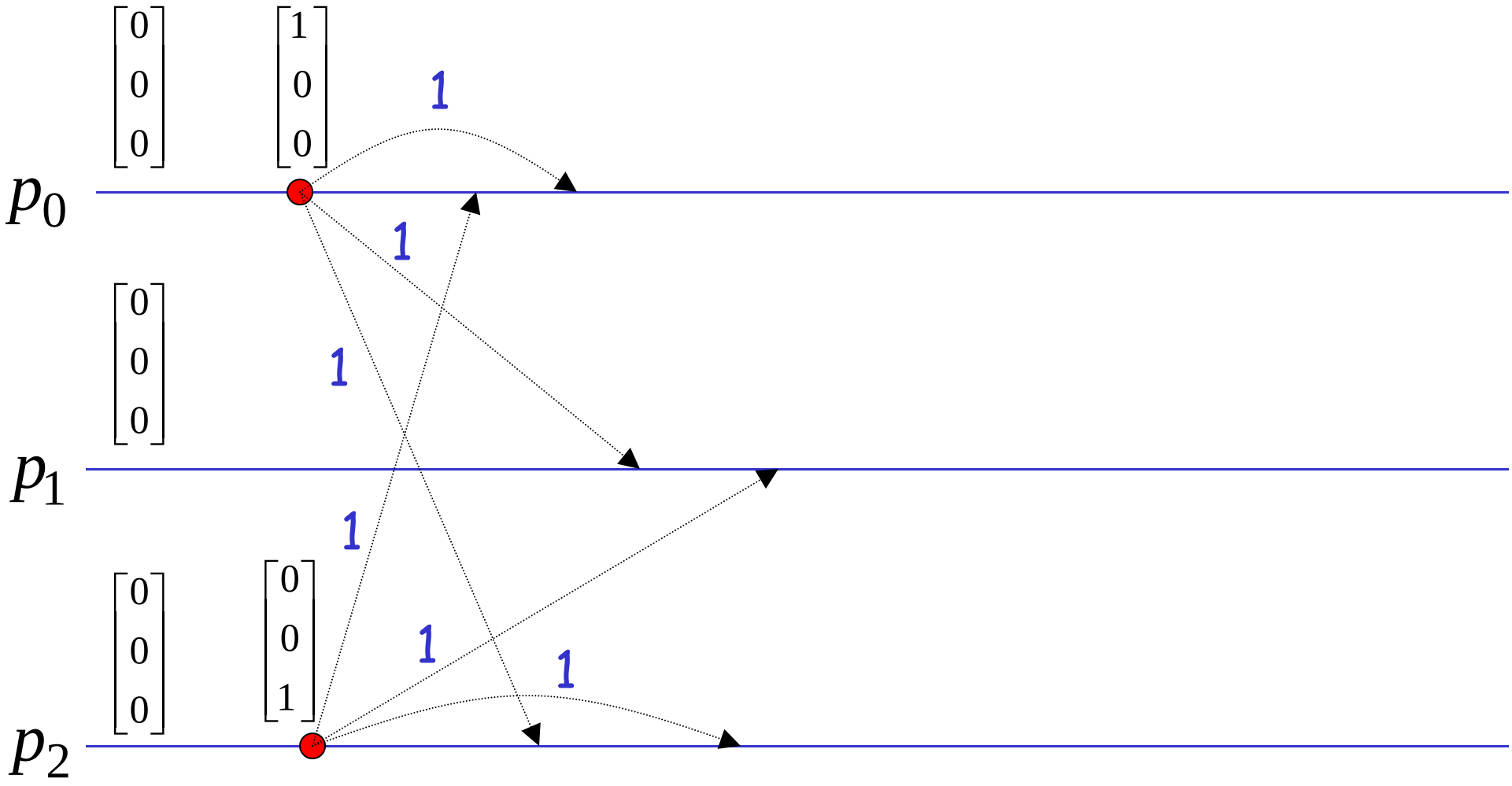
p_0

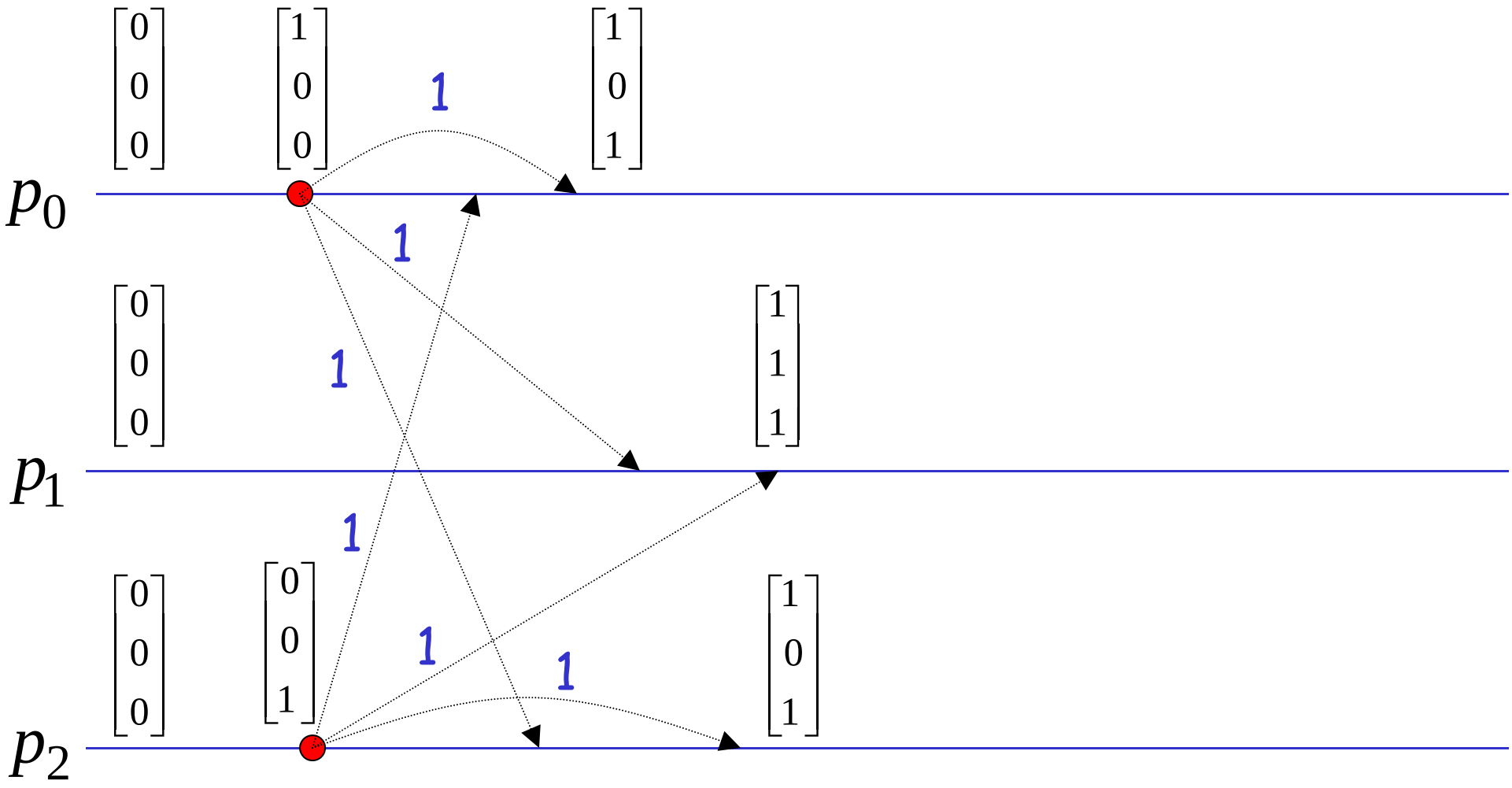
$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

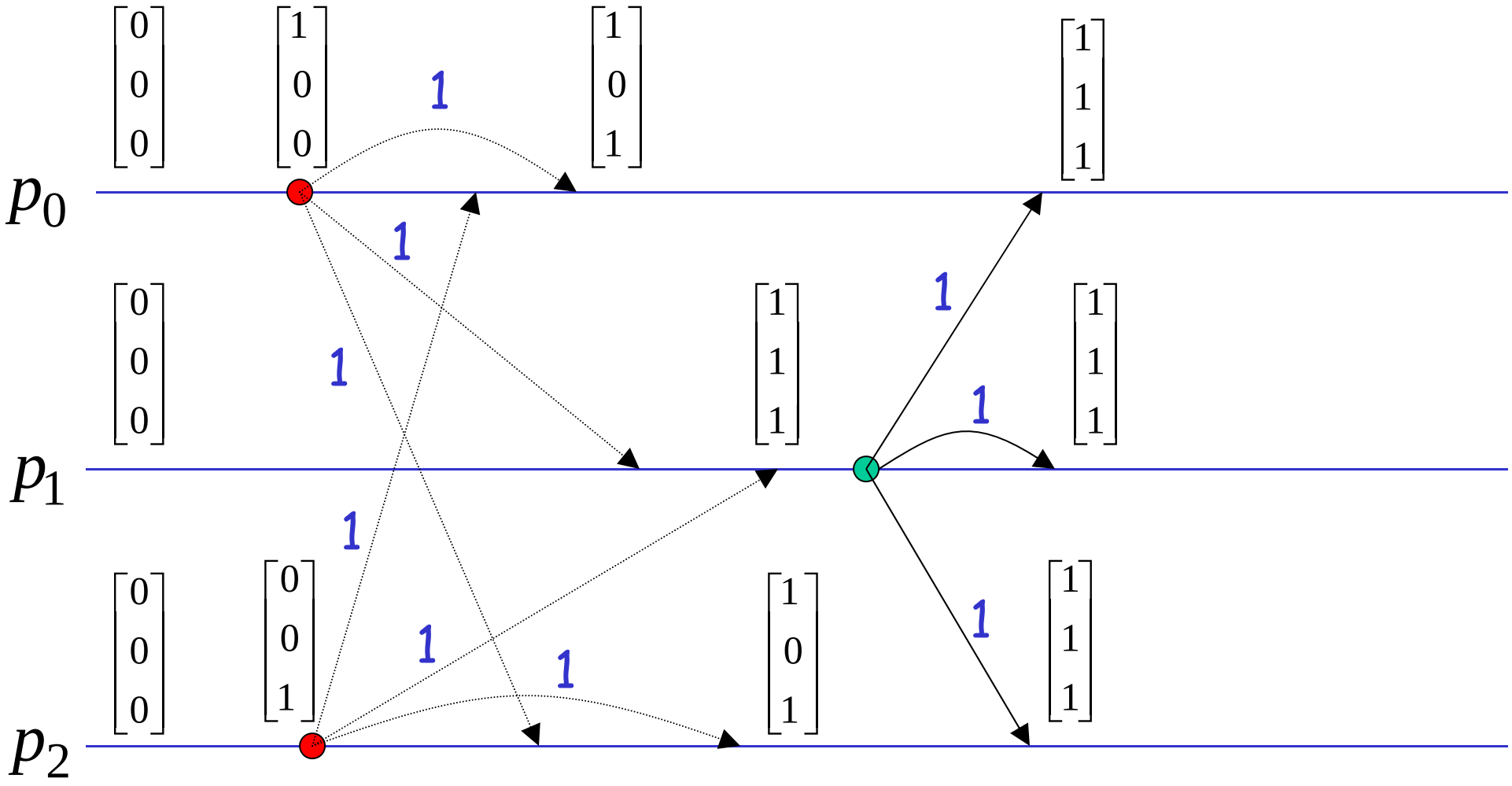
p_1

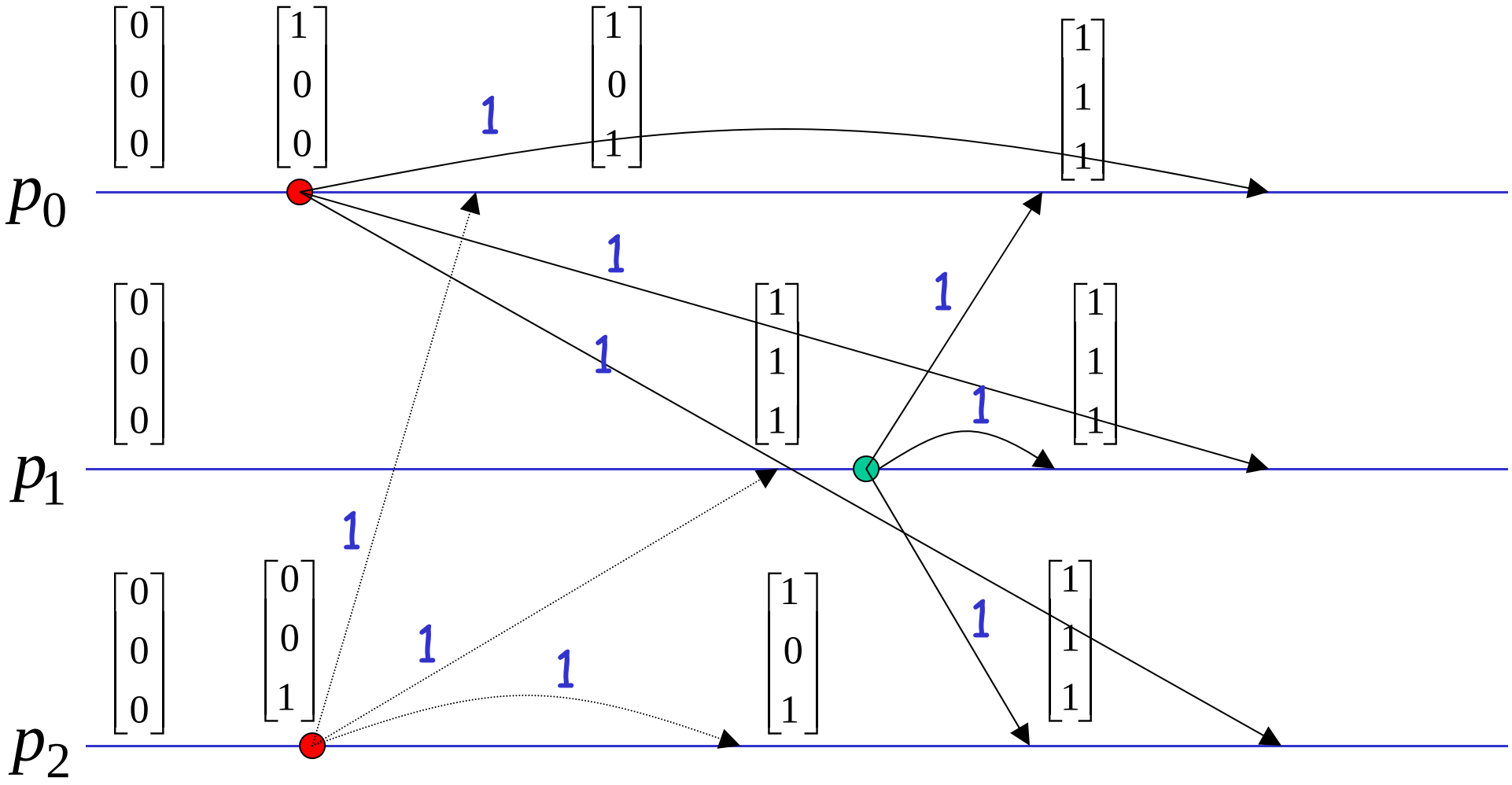
$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

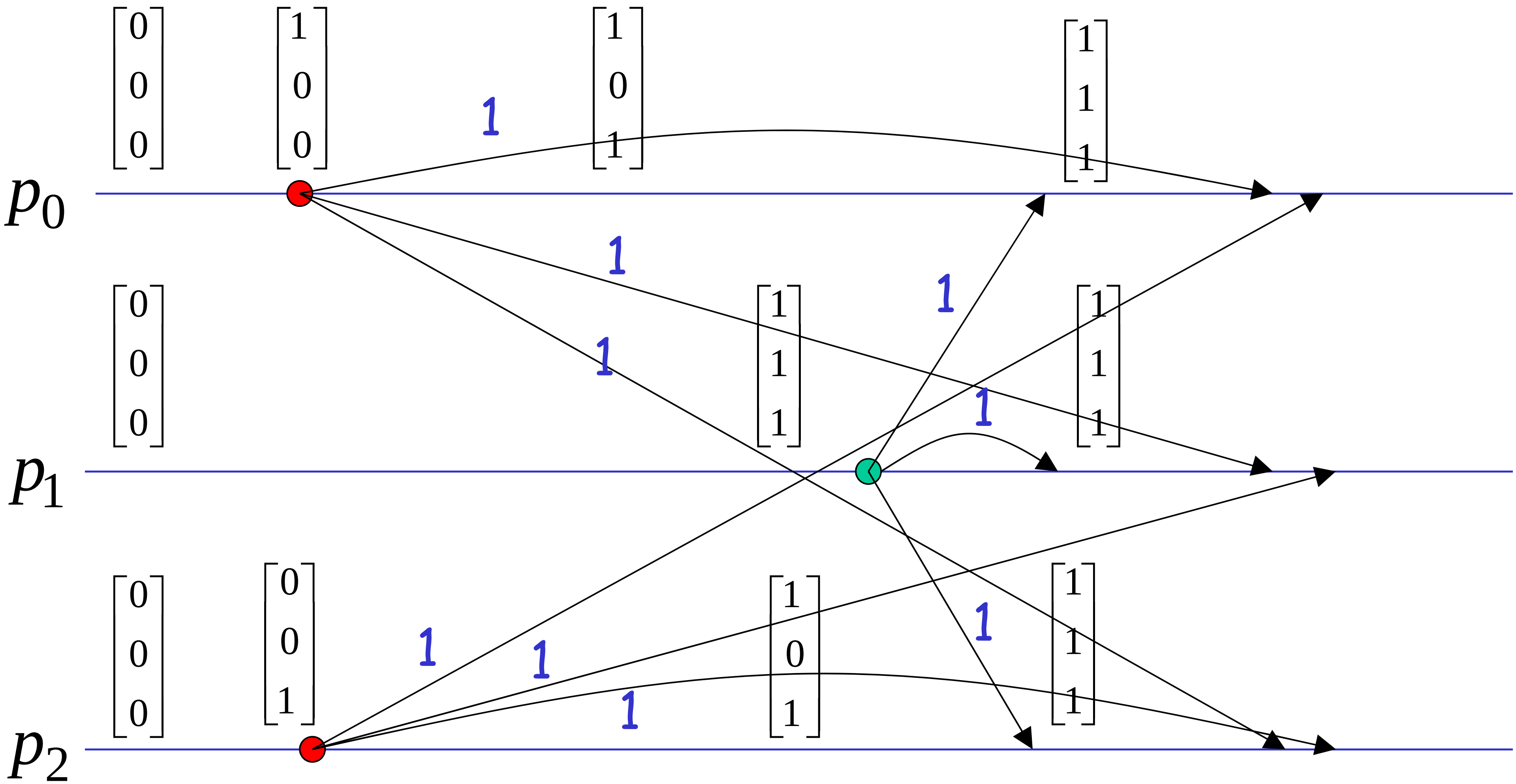
p_2


$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
1
p_0
1
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
1
p_1
1
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
1
1
p_2


p_0
p_1
p_2
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
1
1
1
1
1
1


p_0
p_1
p_2
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\1\\1\end{bmatrix}$
1
1
1
1
1
1
1
1
1
1

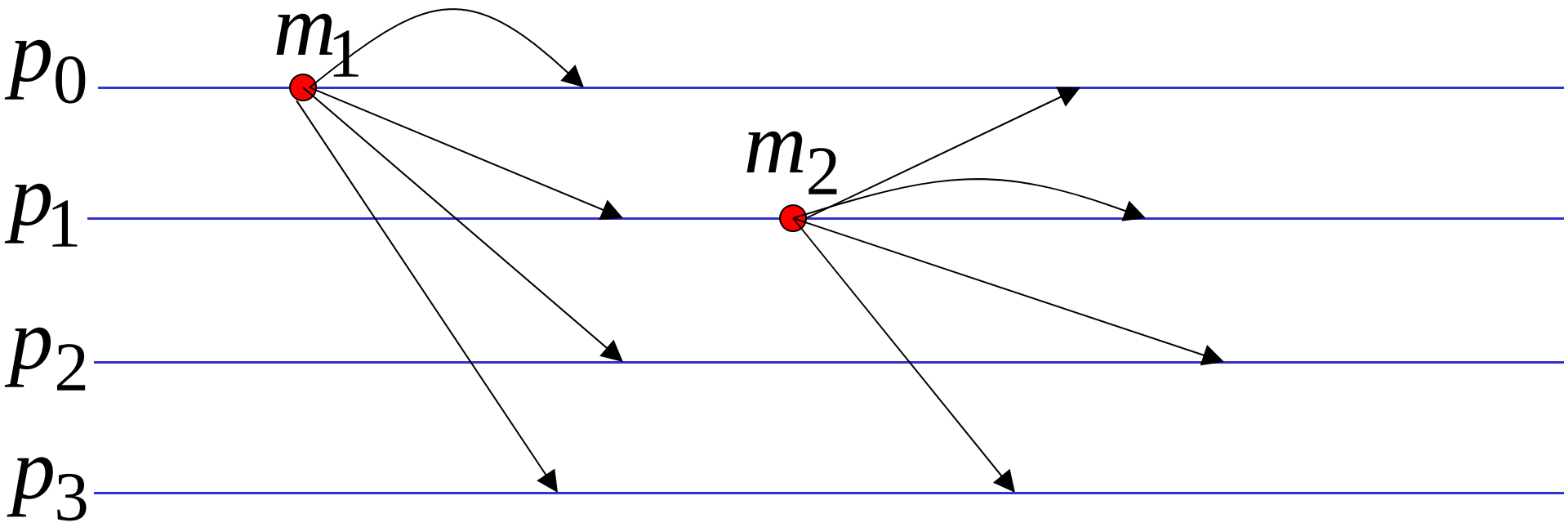

p_0
p_1
p_2
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
1



Simetrija se razbija na osnovu identifikacija čvorova

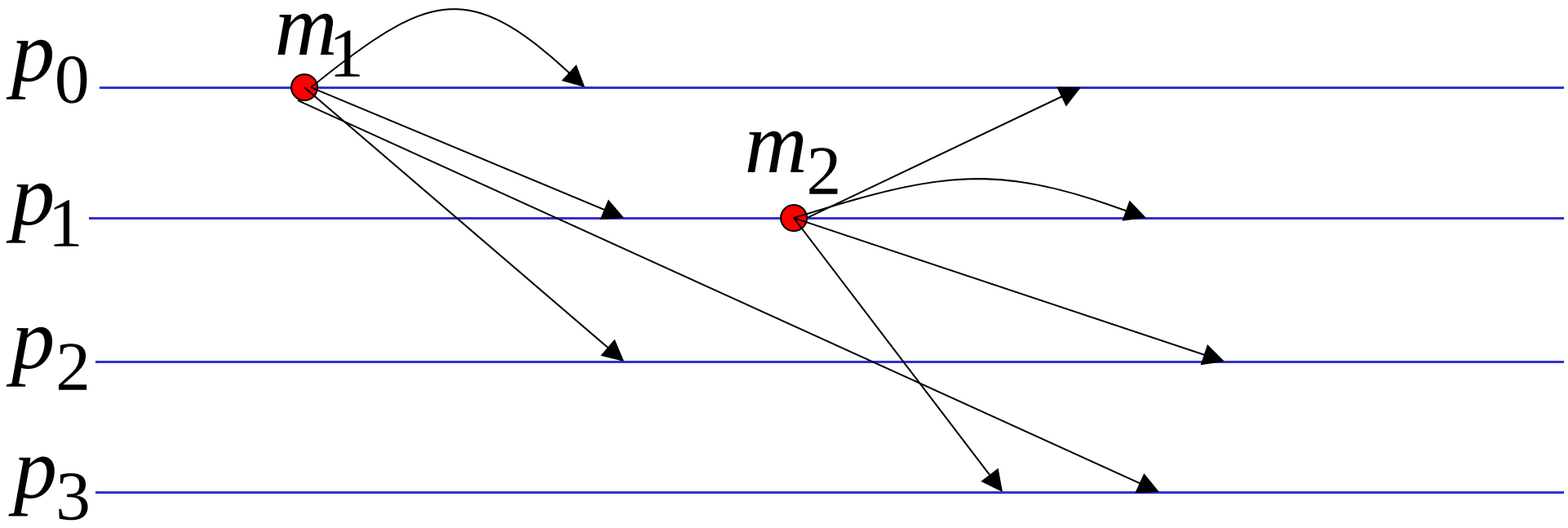
Uzročni redosled – Slanje svima

Uzročni redosled

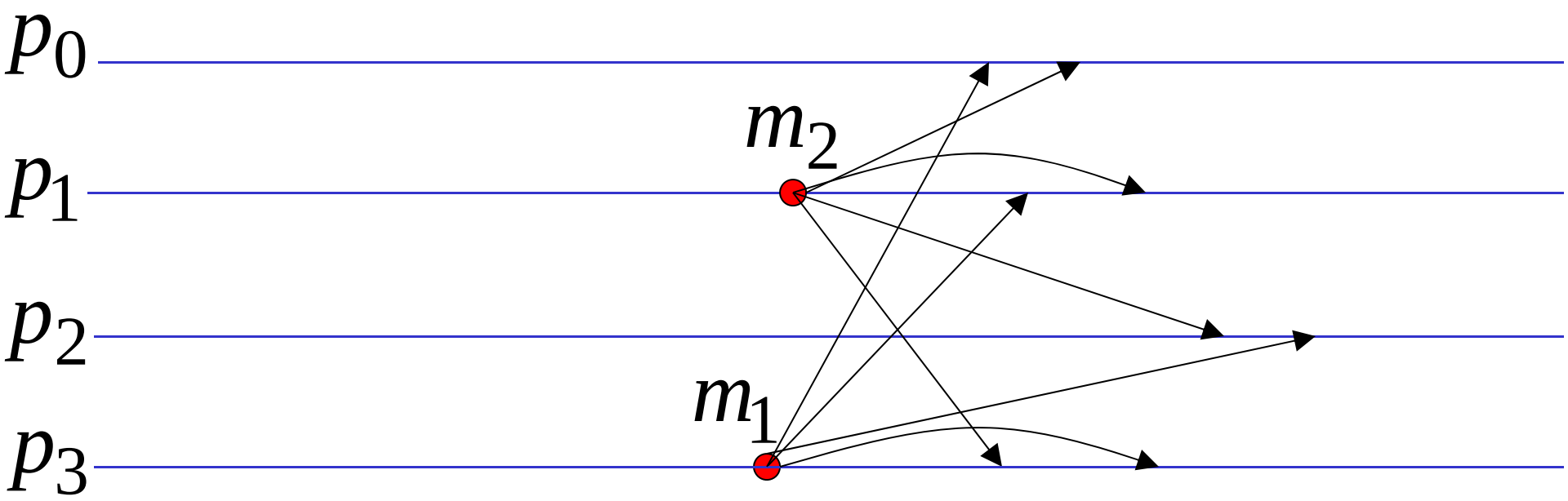


Poruke se primaju u
u redosledu u kom su uzrokovane

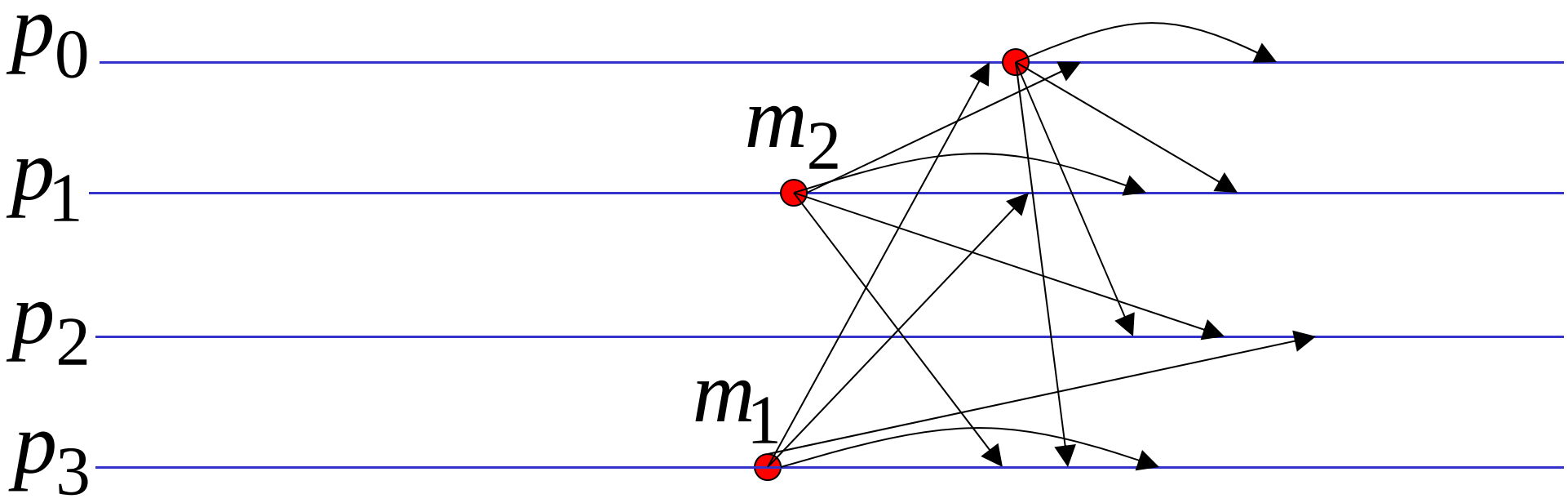
Nije uzročni redosled



Uzročni redosled

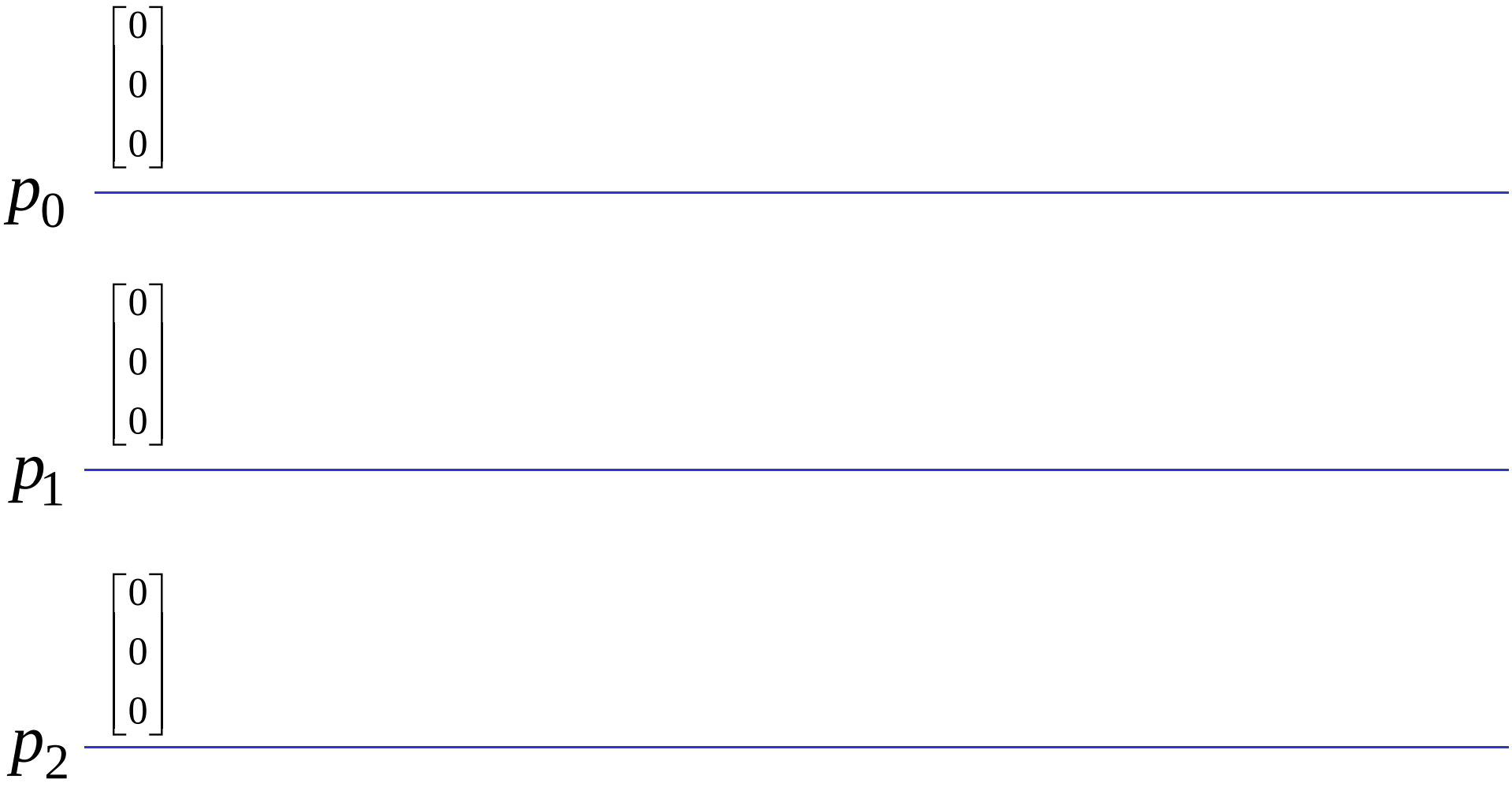


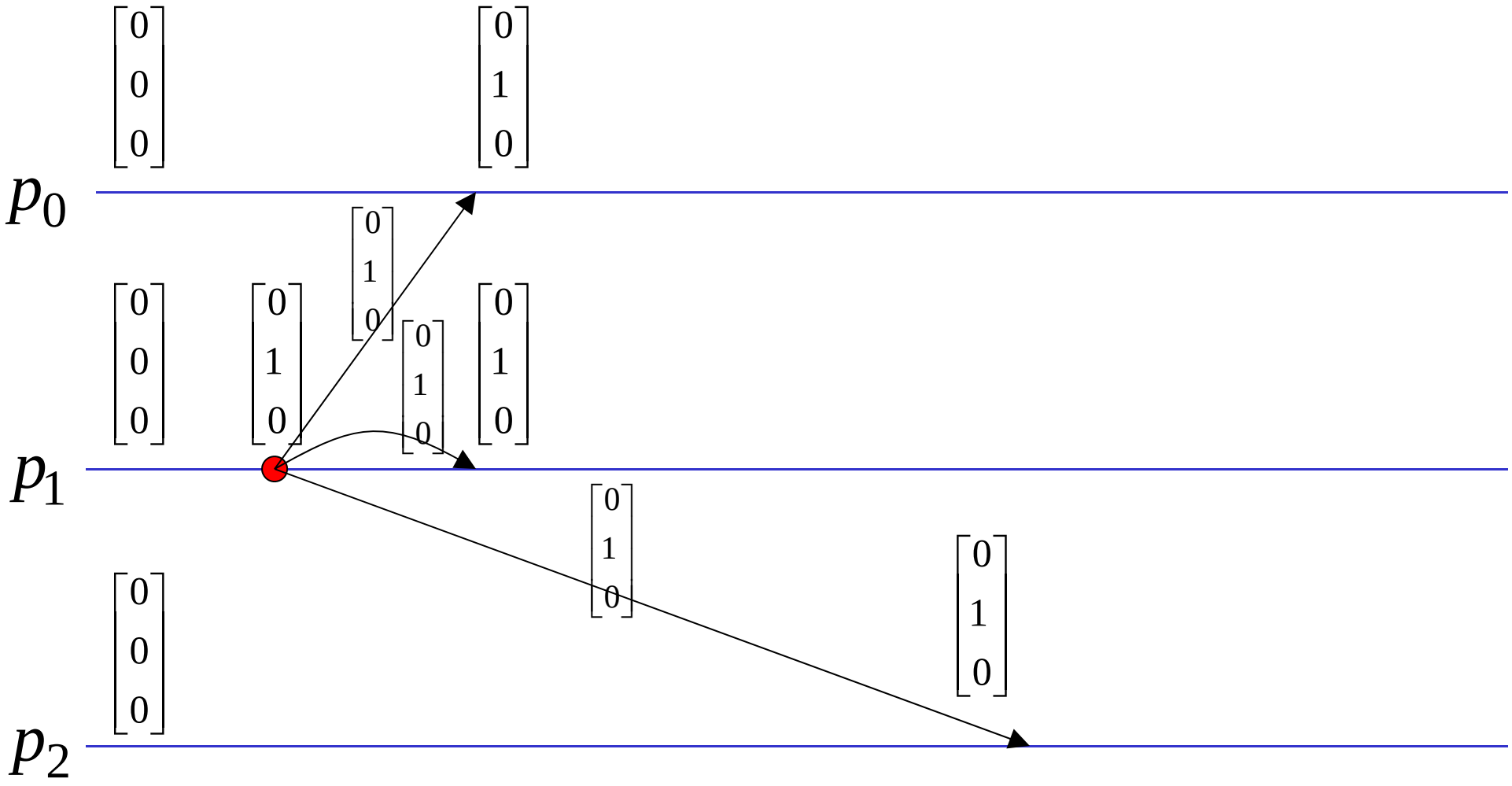
Nije uzročni redosled

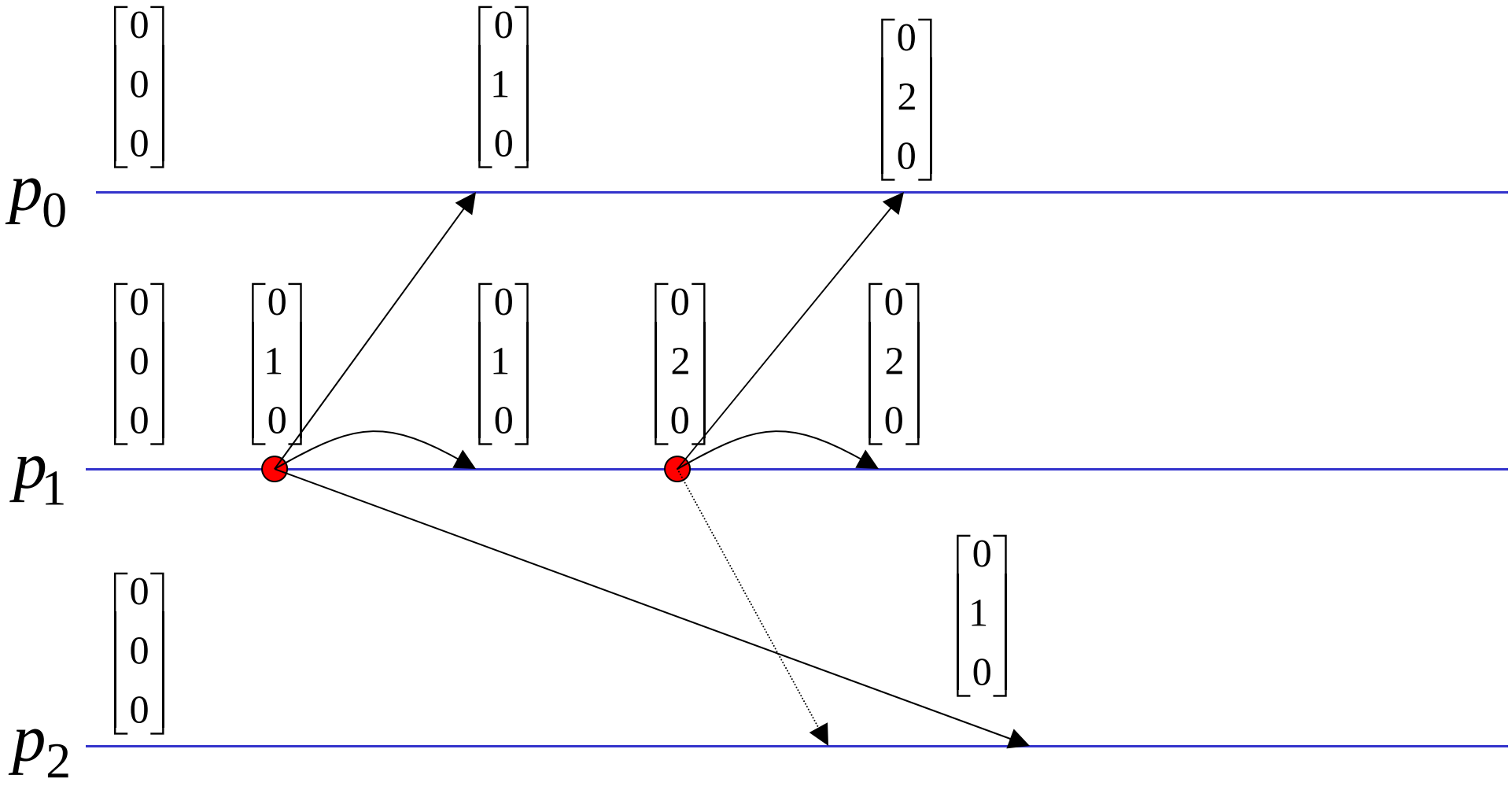


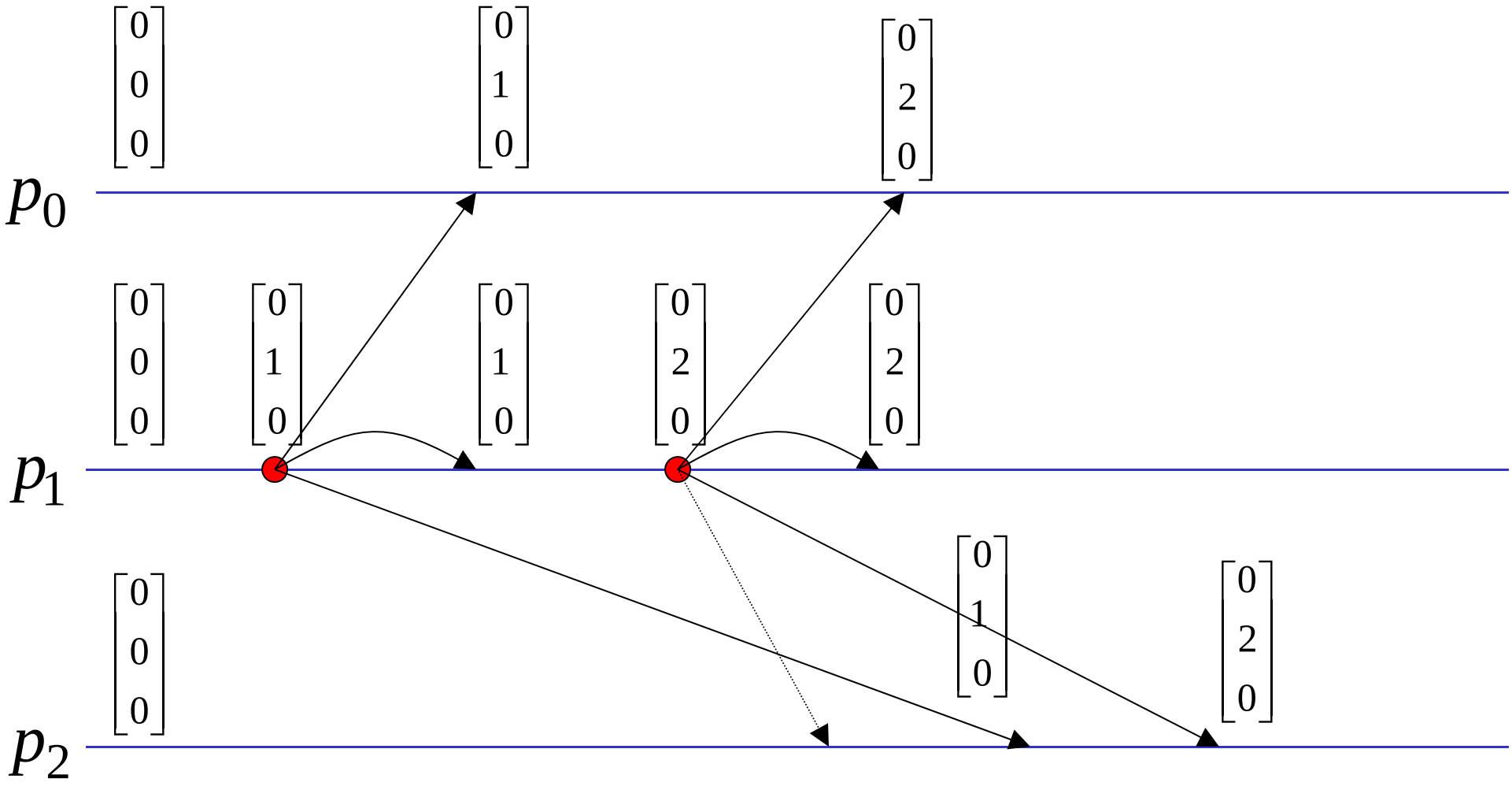
Opservacija: Algoritmi totalnog redosleda koje smo opisali su takođe algoritmi uzročnog redosleda

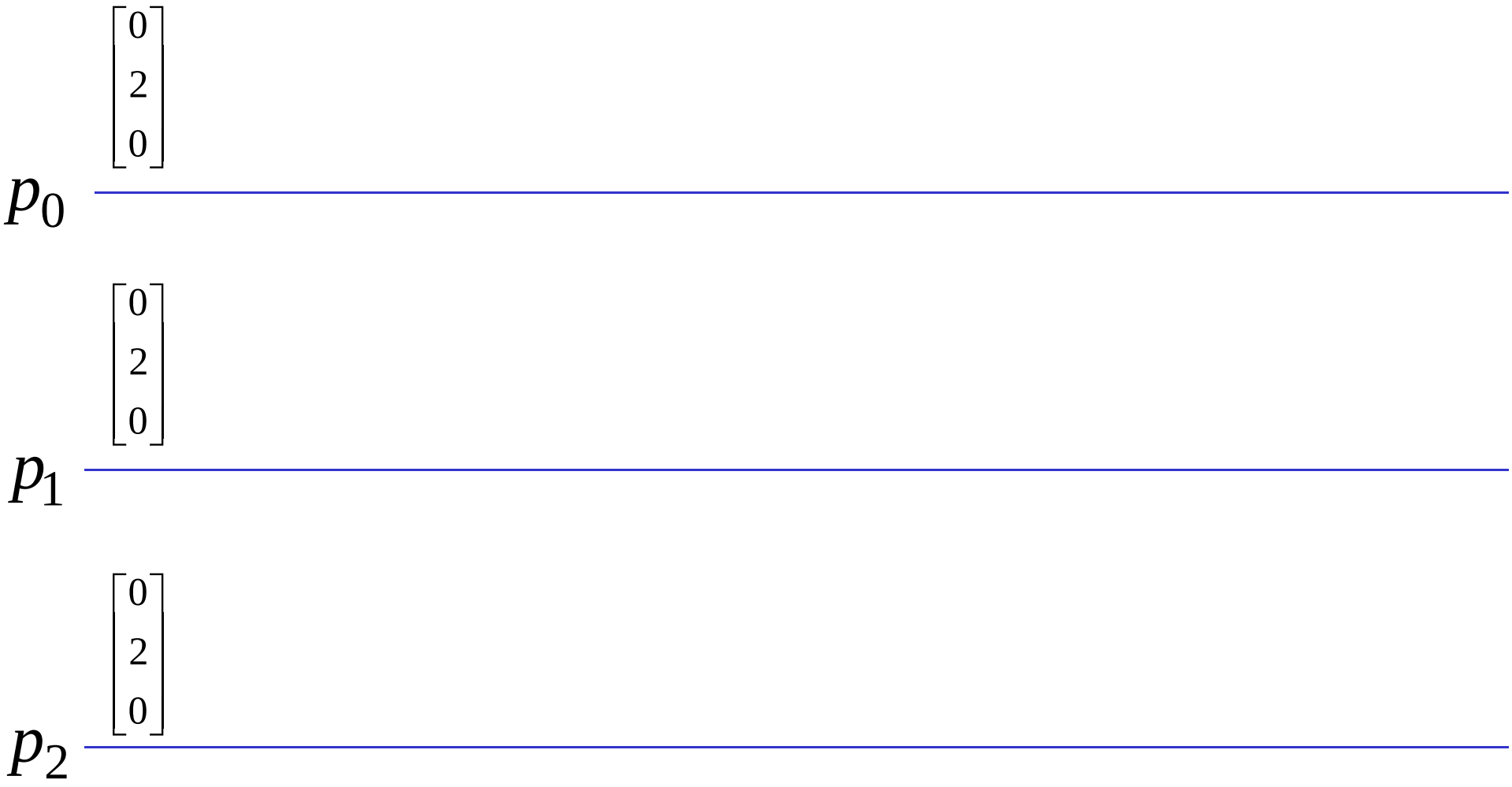
Postoji efikasniji algoritam u smislu broja poruka

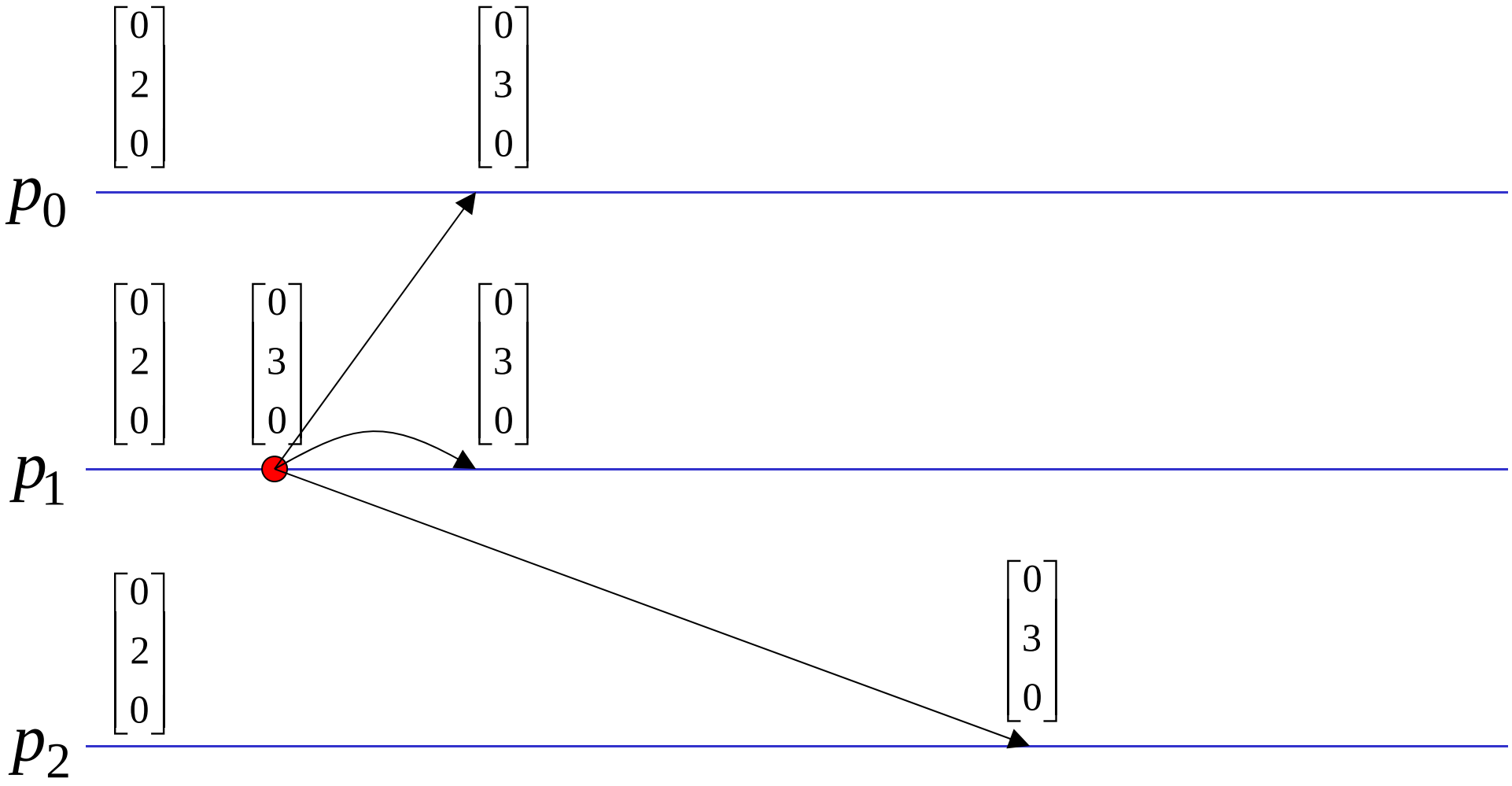

$\begin{bmatrix}0\\0\\0\end{bmatrix}$
p_0
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
p_1
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
p_2

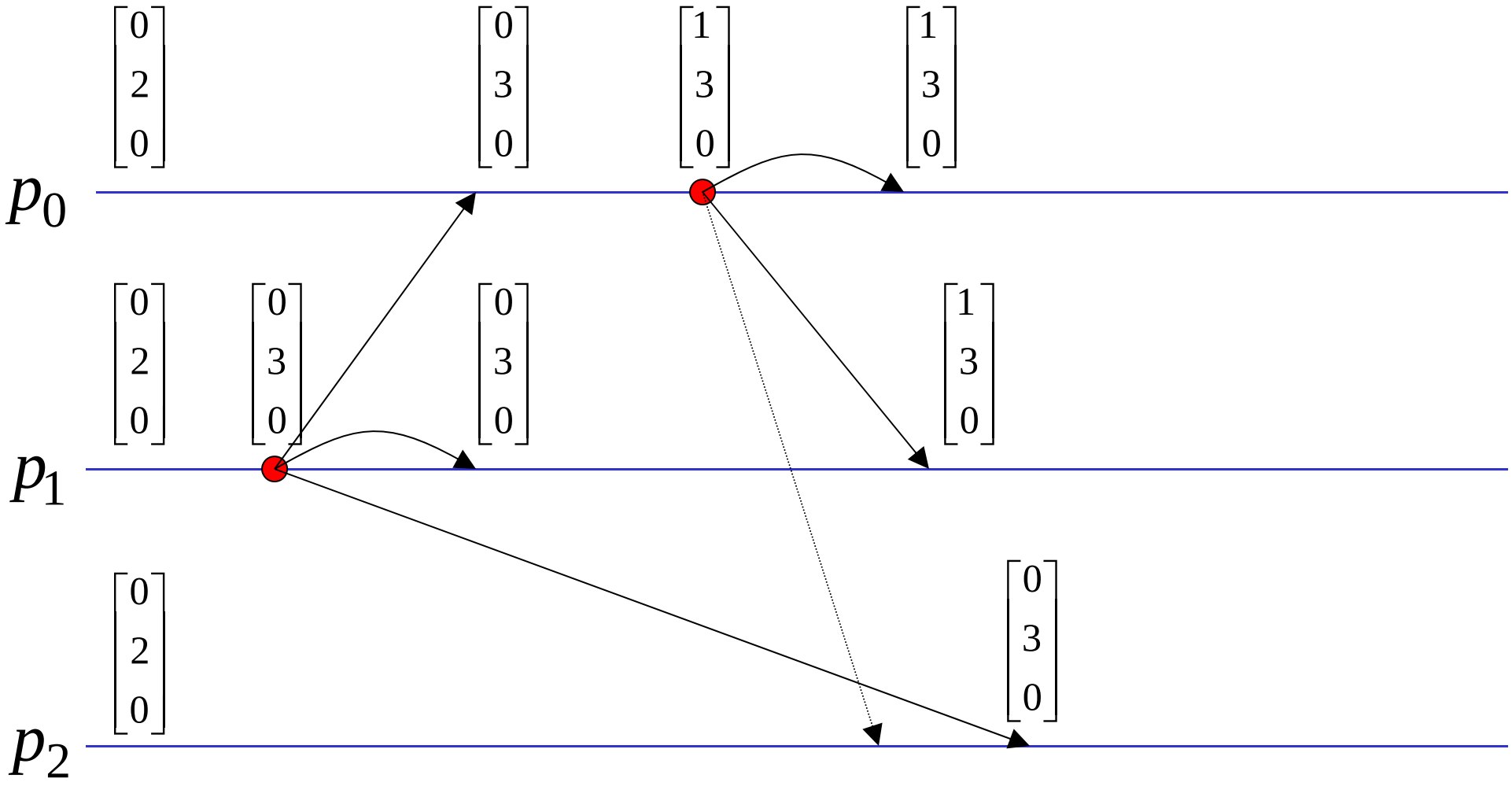

p_0
p_1
p_2
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$

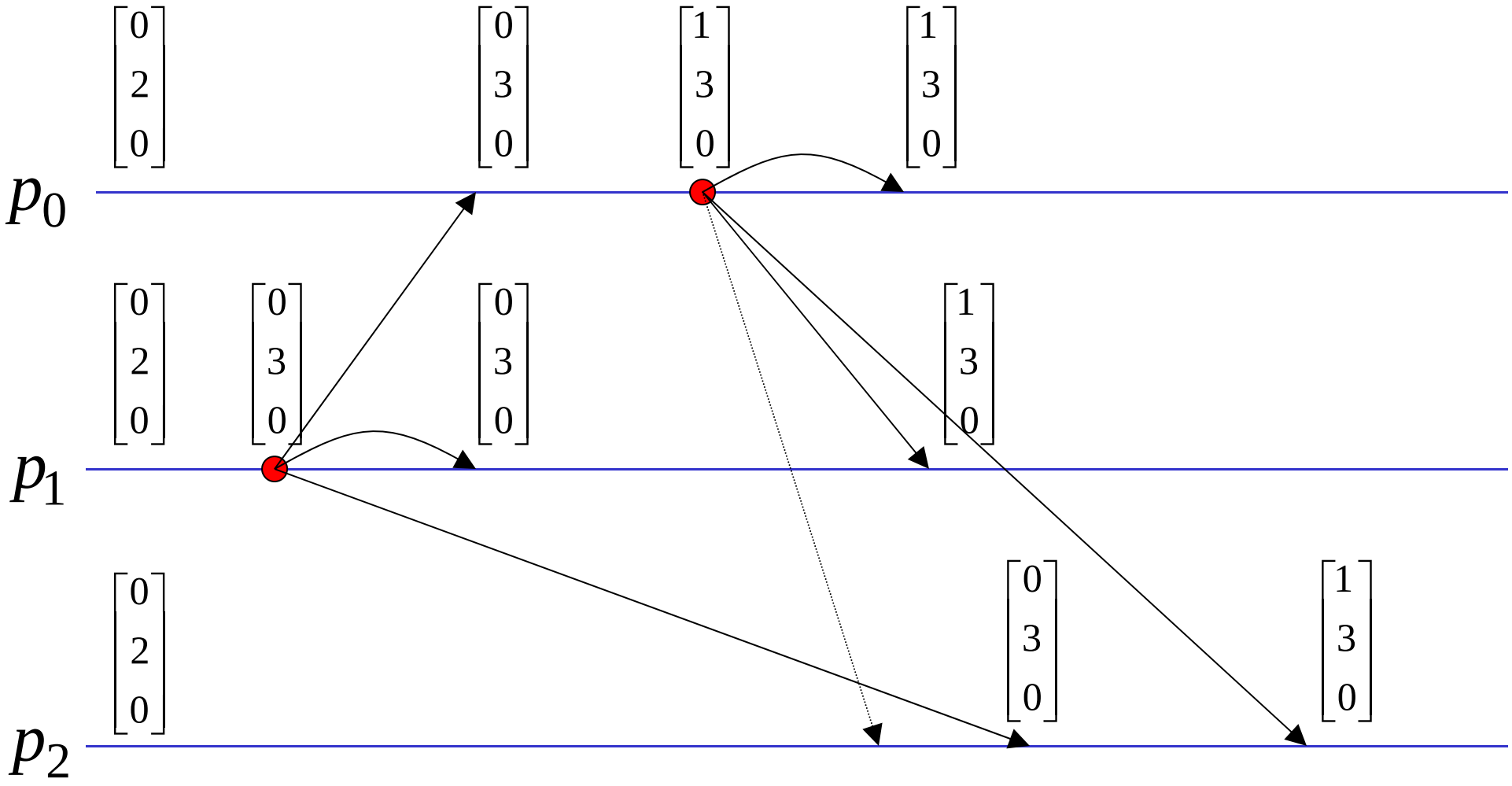

p_0
p_1
p_2
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$


p_0
p_1
p_2
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$


$\begin{bmatrix}0\\2\\0\end{bmatrix}$
p_0
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
p_1
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
p_2


$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
p_0
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
p_1
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
p_2


p_0
p_1
p_2
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}1\\3\\0\end{bmatrix}$
$\begin{bmatrix}1\\3\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}1\\3\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$


p_0
p_1
p_2
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}1\\3\\0\end{bmatrix}$
$\begin{bmatrix}1\\3\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}1\\3\\0\end{bmatrix}$
$\begin{bmatrix}0\\2\\0\end{bmatrix}$
$\begin{bmatrix}0\\3\\0\end{bmatrix}$
$\begin{bmatrix}1\\3\\0\end{bmatrix}$

Procesor p_i zvanično prima poruku od p_j sa vektor satom

$$\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \dots \end{bmatrix}$$

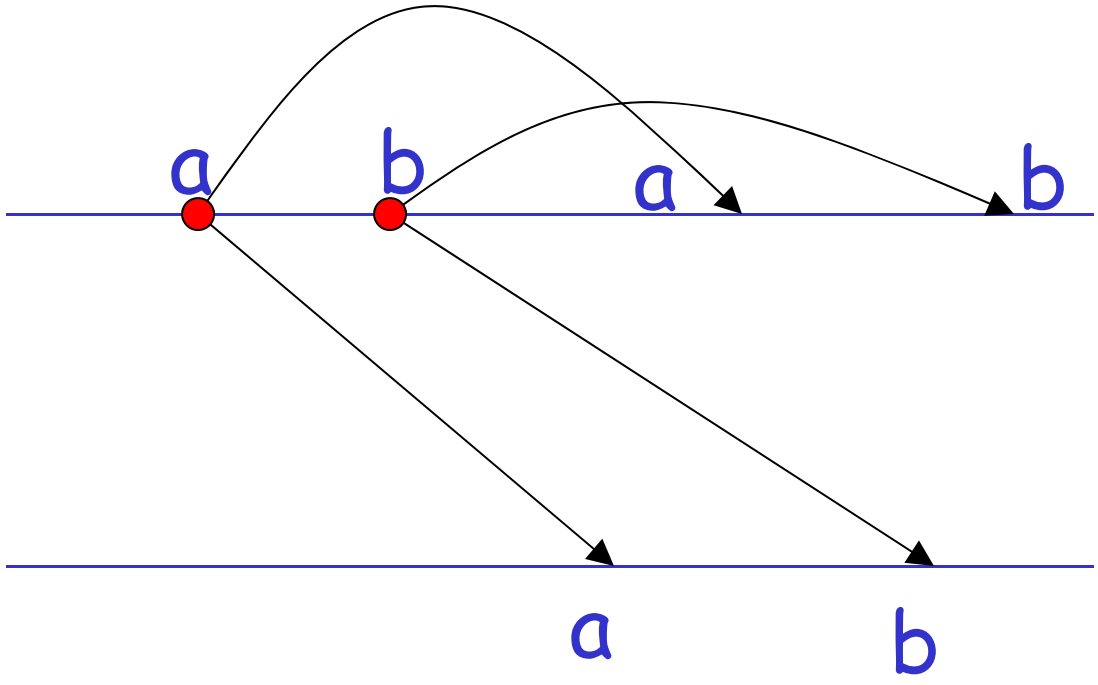
Kada:

1) Sve poruke od p_k $(k \neq j)$ manje ili jednake sa v_k stignu

2) Sve poruke od p_j manje od v_j stignu

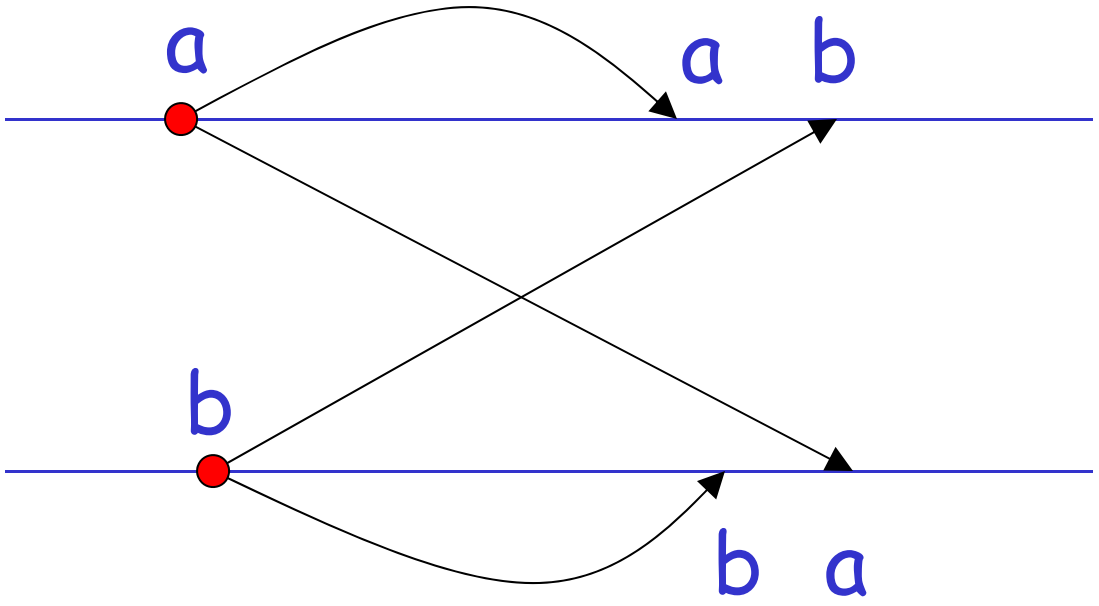
Relacije između raznih redosleda zasnovanih na slanju svima

Uzročni redosled



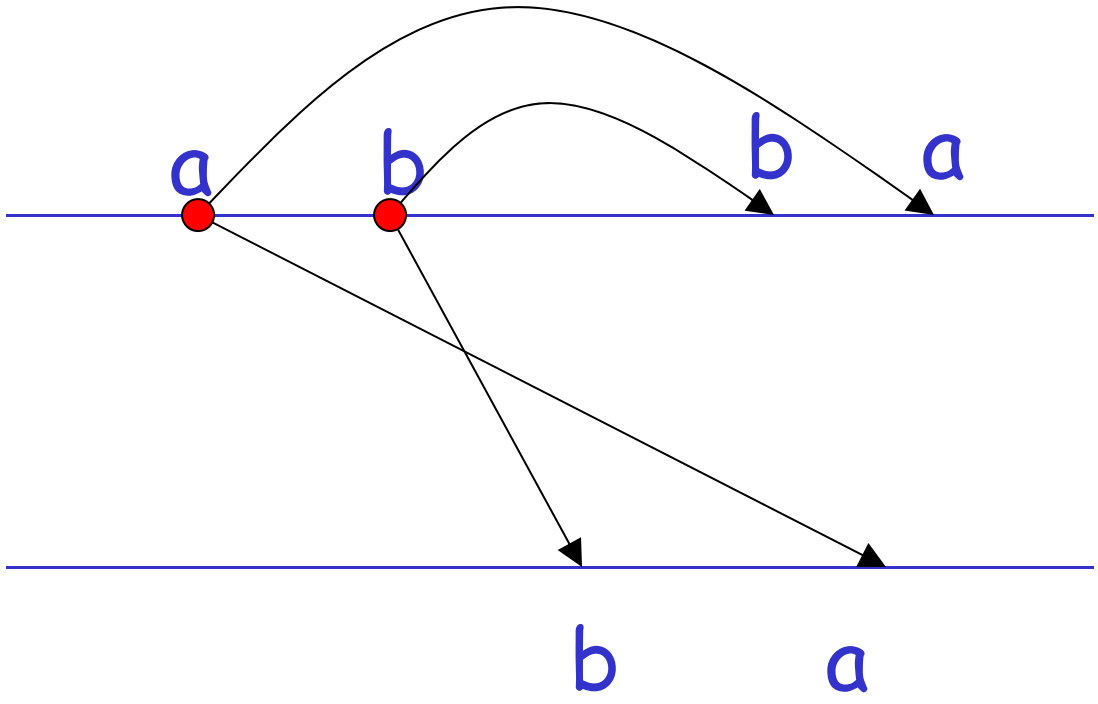
Uzročni redosled implicira FIFO por. iz jednog izvora

Uzročni redosled



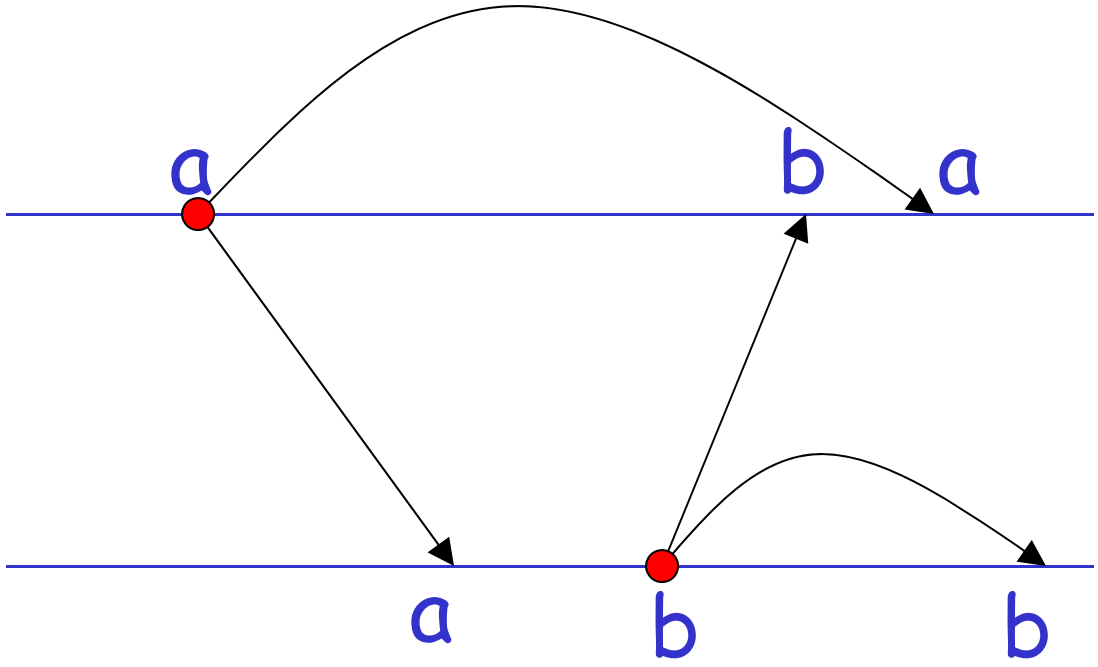
Uzročni redosled ne implicira totalni redosled

Totalni redosled



Totalni redosled ne implicira uzročni redosled niti FIFO por. iz jednog izvora

FIFO por. iz jednog izvora



FIFO por. iz jednog izvora ne implicira uzročni redosled niti totalni redosled