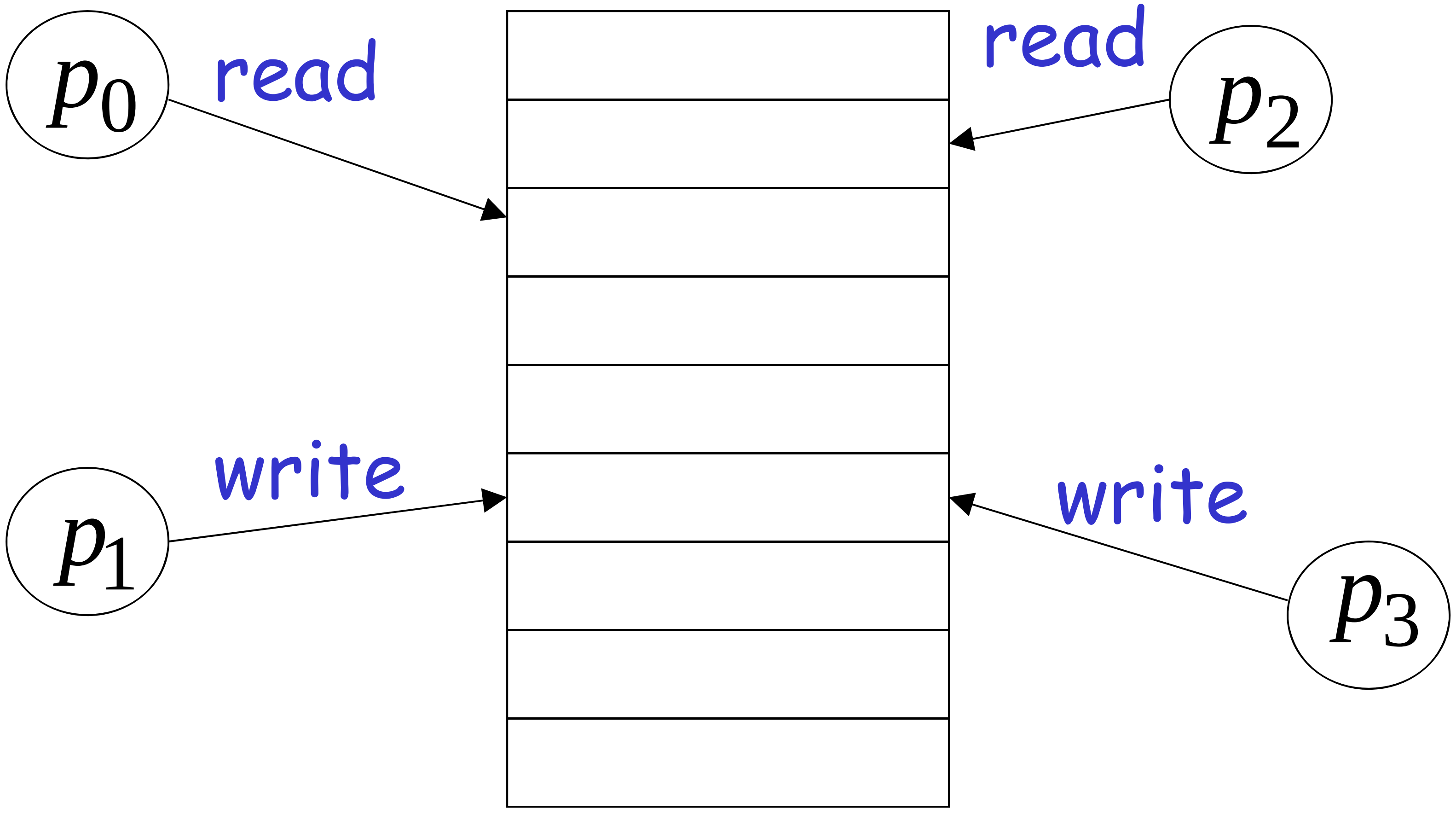
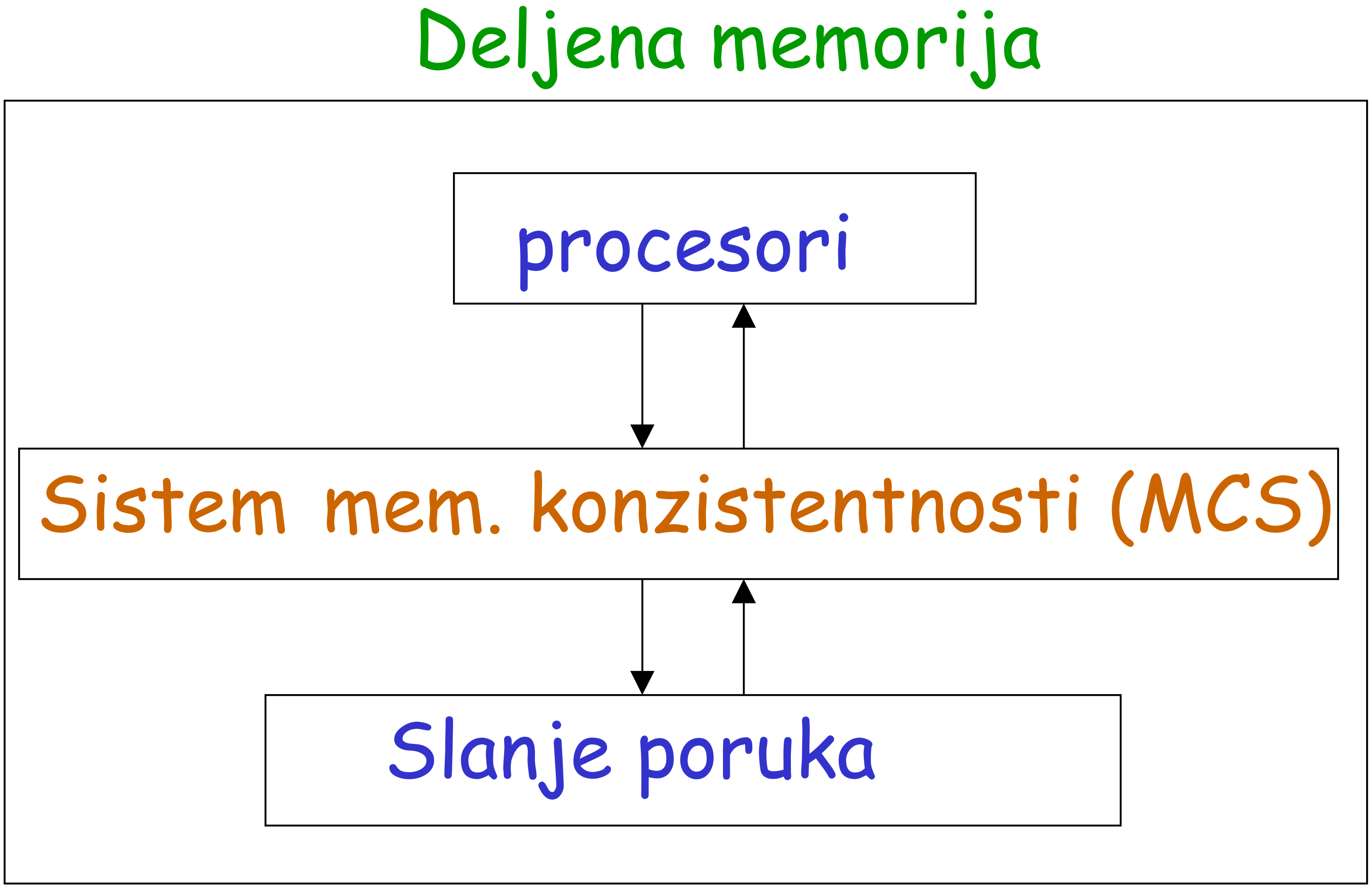
Sistemi konzistentne deljene memorije

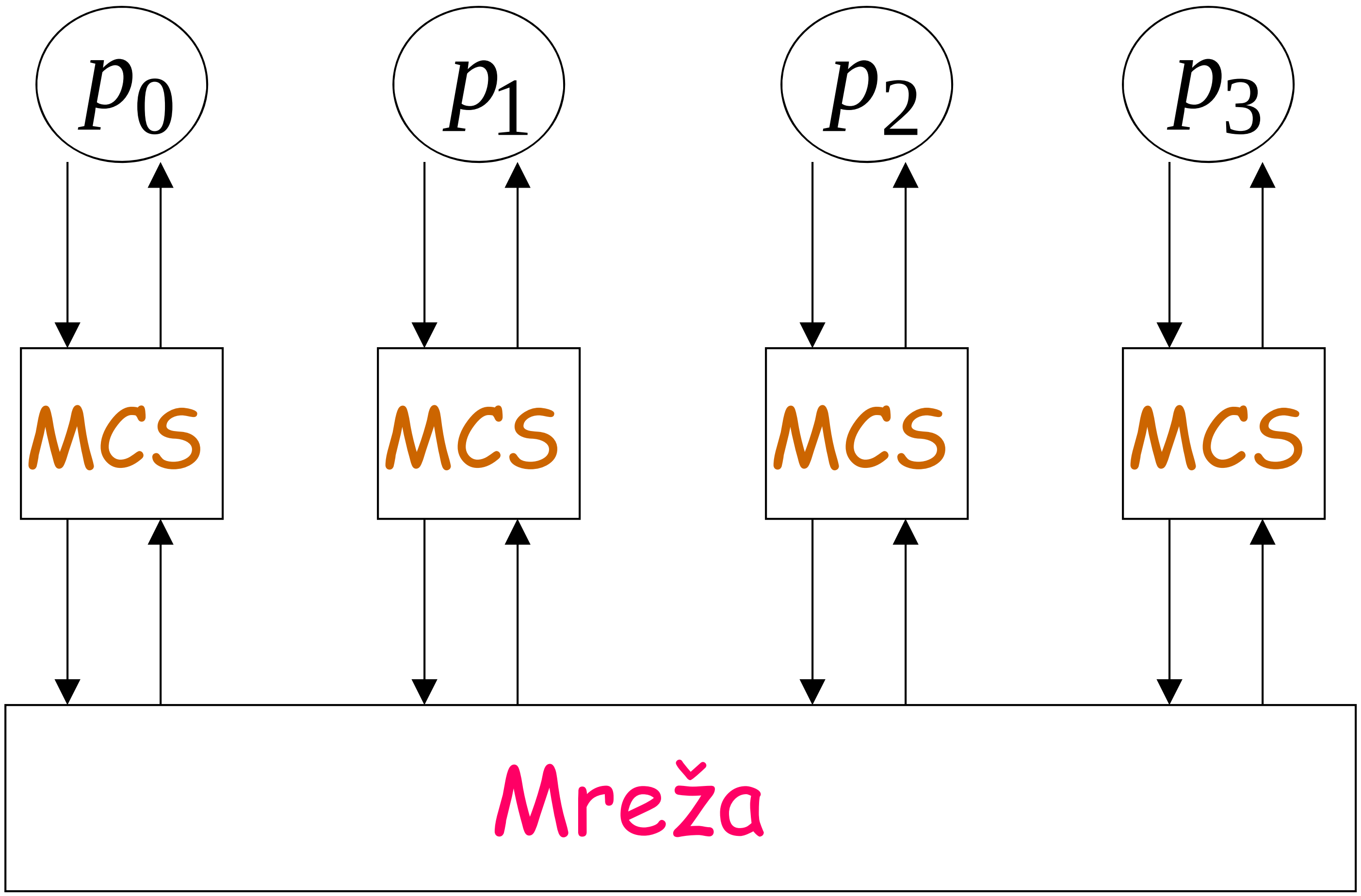
Deljena memorija





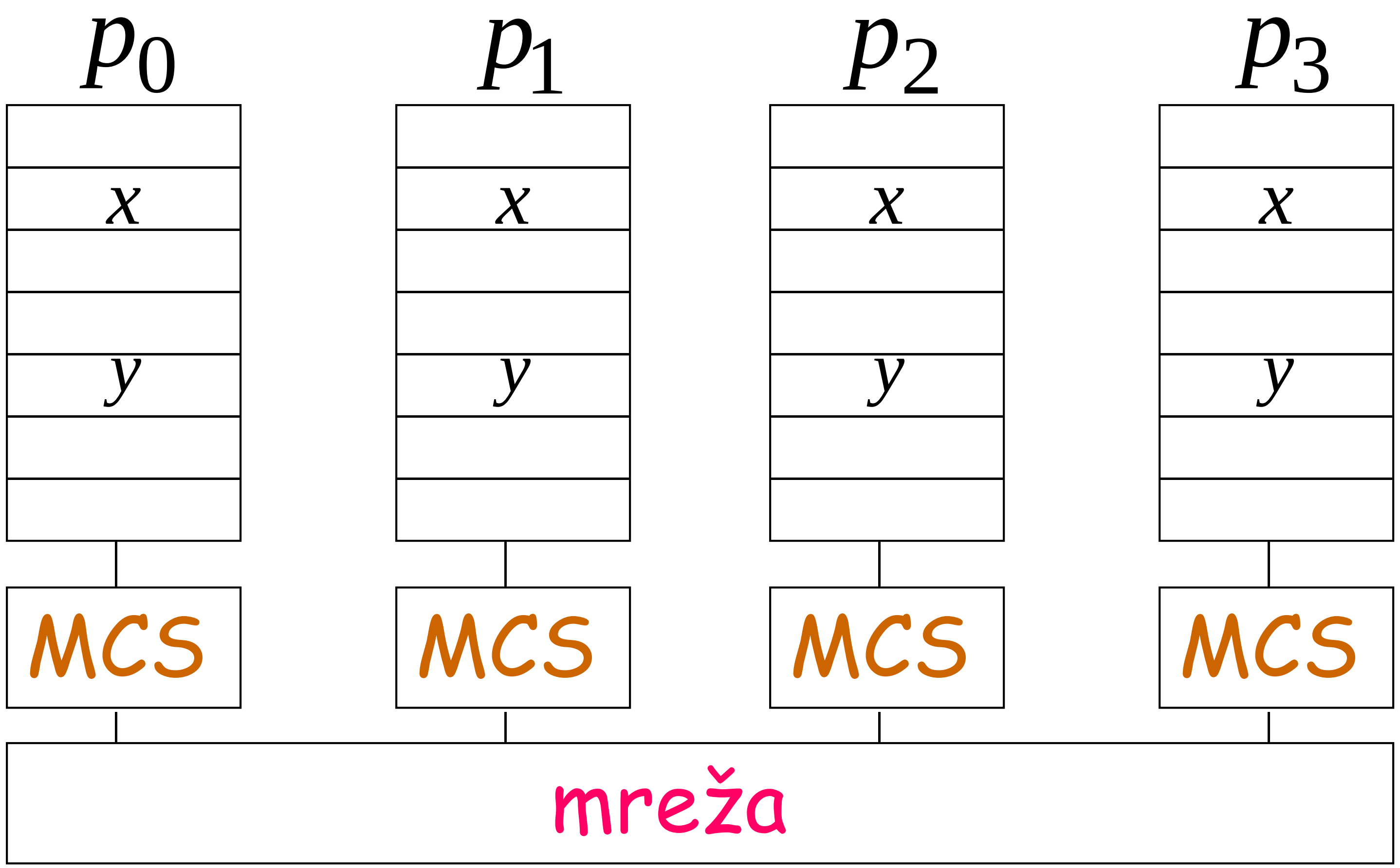
MCS(Memory Consistency System)

Deljena memorija,
na osnovi za slajnje poruka



Moguća implementacija:

Svaki procesor ima svoju sopstvenu kopiju deljene memorije

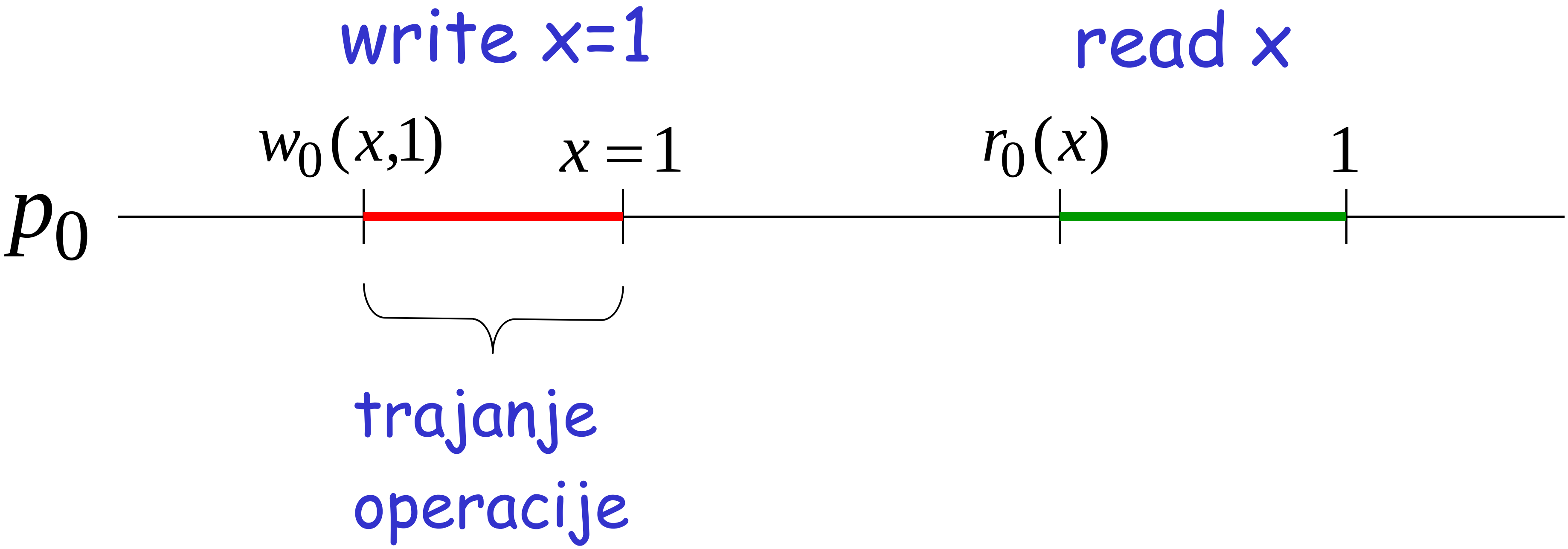


Sistem memorijske konzistentnosti održava konzistentnost deljene memorije

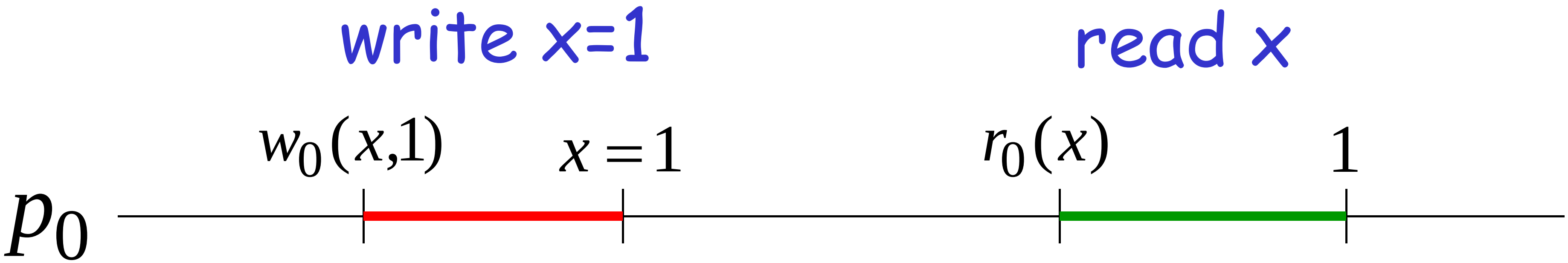
Svaki procesor ima isti pogled na deljenu memoriju

Izvršenja

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

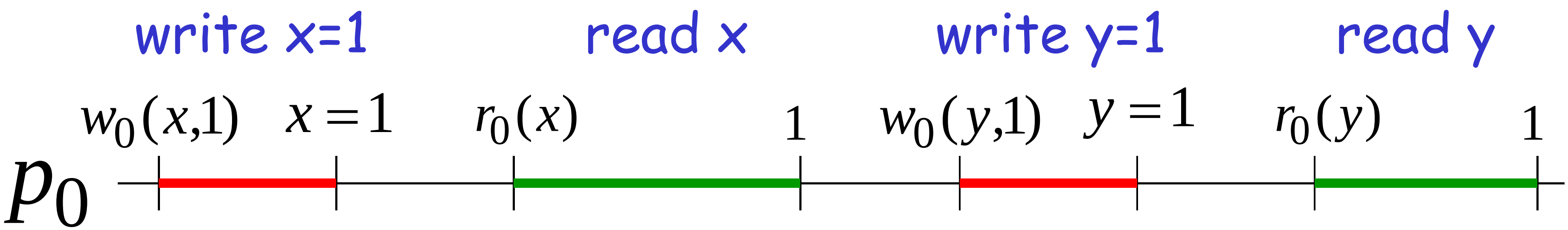


$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$



Sekvencijalni raspored: $w_0(x,1), r_0(x)$

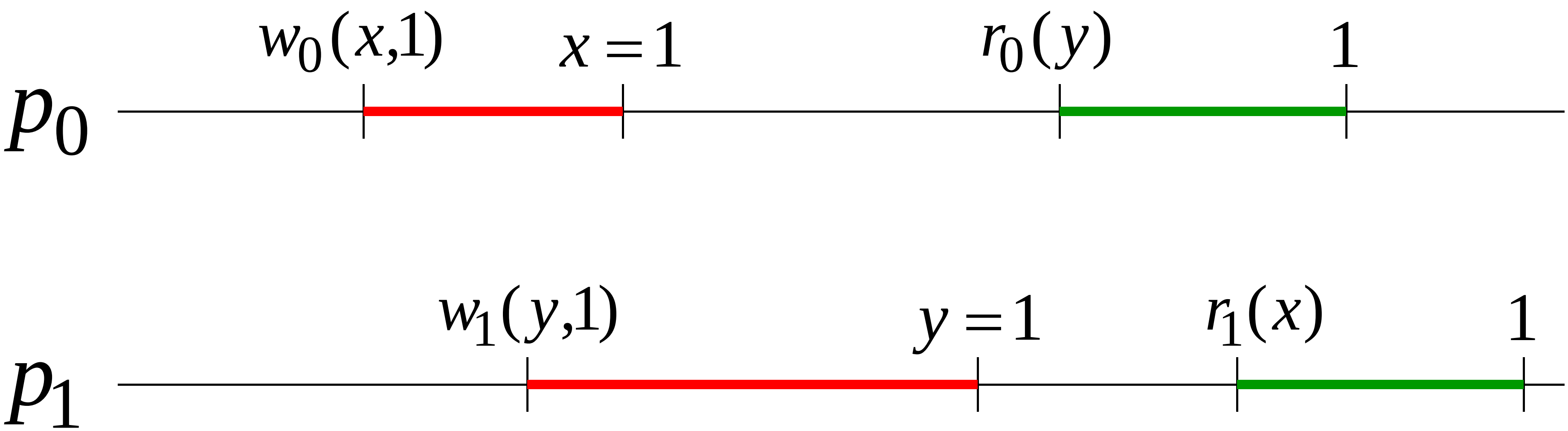
$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$



sekvencijalni raspored:

$$w_0(x,1),\ r_0(x),\ w_0(y,1),\ r_0(y)$$

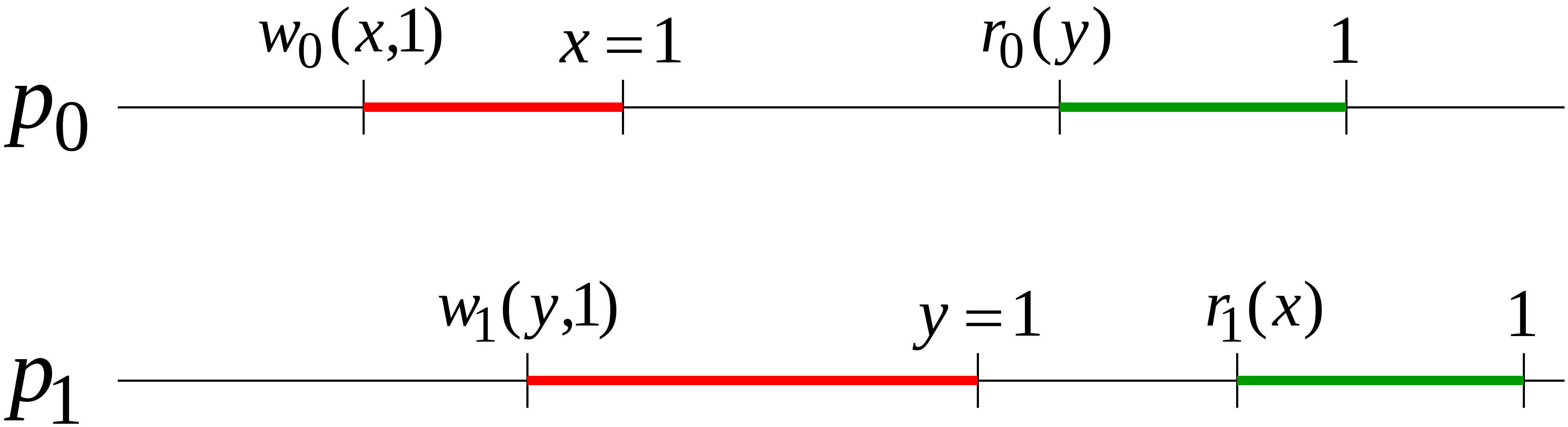
$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$



Paralelni događaji: $w_0(x,1) \;//\; w_1(y,1)$

$$r_0(y) \;//\; r_1(x)$$

$$\boxed{\begin{matrix} x = 0 \\ y = 0 \end{matrix}}$$

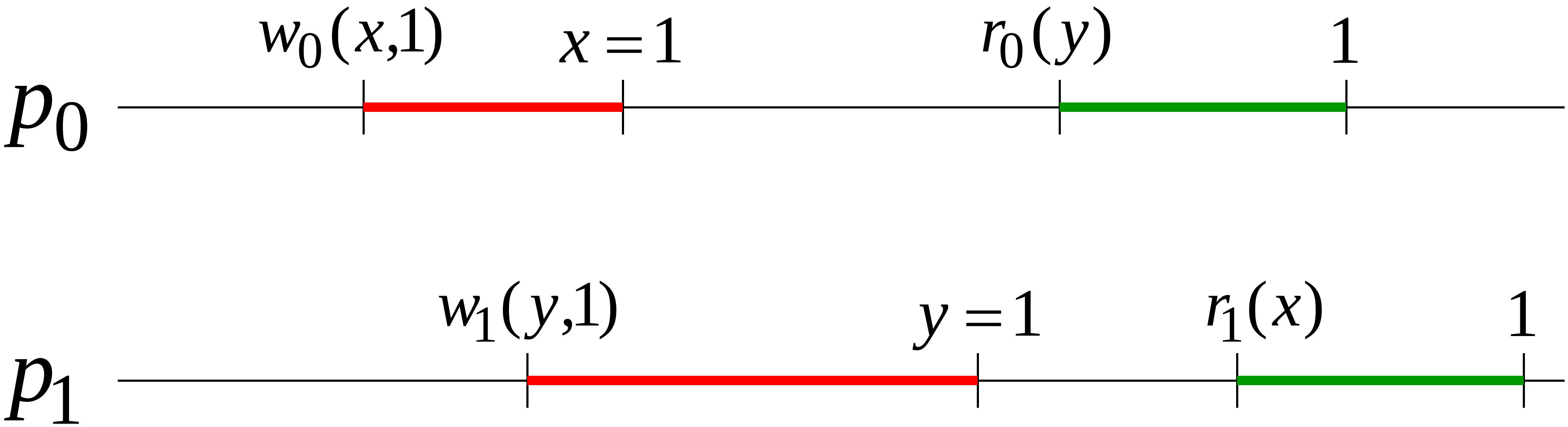


Uzročni događaji:

$w_0(x,1) \; < \; r_0(y)$ $\qquad\qquad$ $w_0(x,1) \; < \; r_1(x)$

(isti procesor) $\qquad\qquad$ (validne operacije)

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

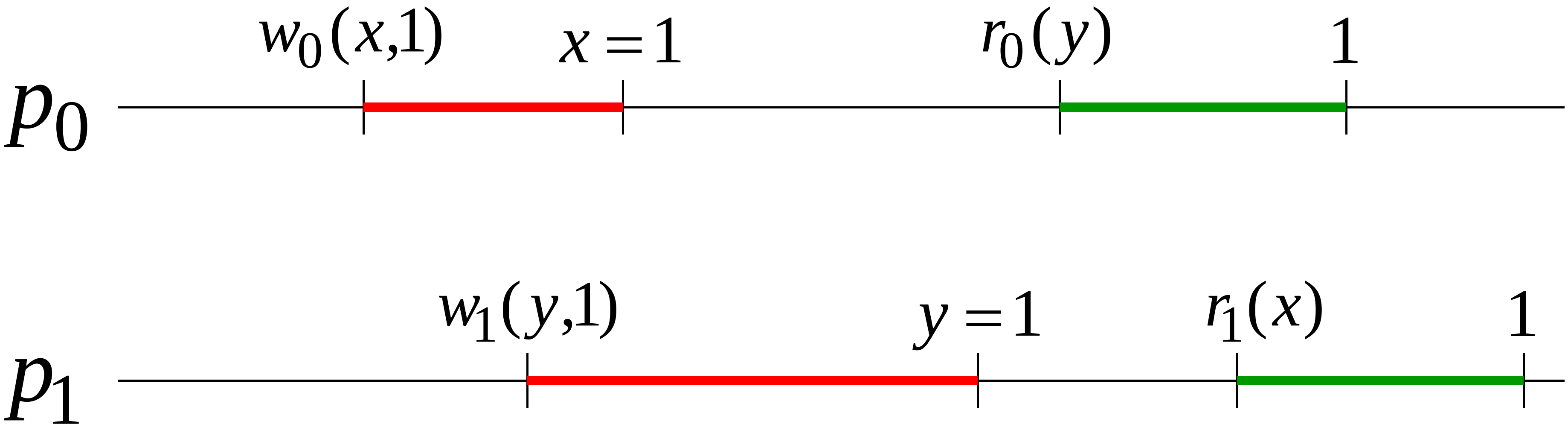


Svi uzročni događaji:

$$w_0(x,1) < r_0(y) \qquad w_1(y,1) < r_0(y)$$

$$w_0(x,1) < r_1(x) \qquad w_1(y,1) < r_1(x)$$

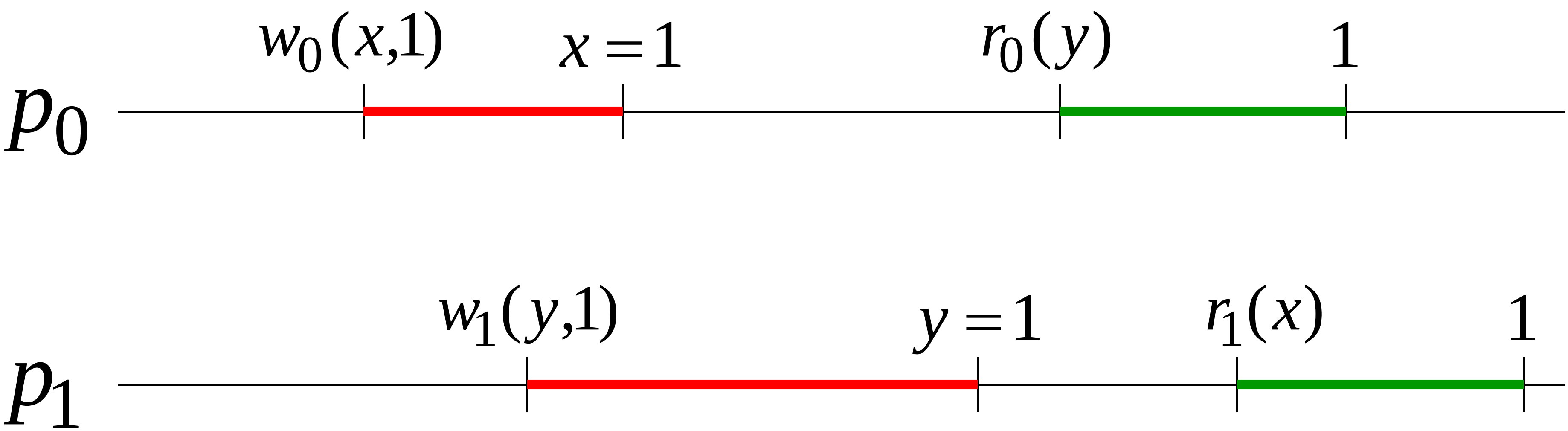
$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$



Validna sekvencijalna izvršenja:
očuvavaju uzročnost događaja

$w_0\, w_1\, r_0\, r_1 \qquad w_1\, w_0\, r_0\, r_1 \qquad w_0\, w_1\, r_1\, r_0 \qquad w_1\, w_0\, r_1\, r_0$

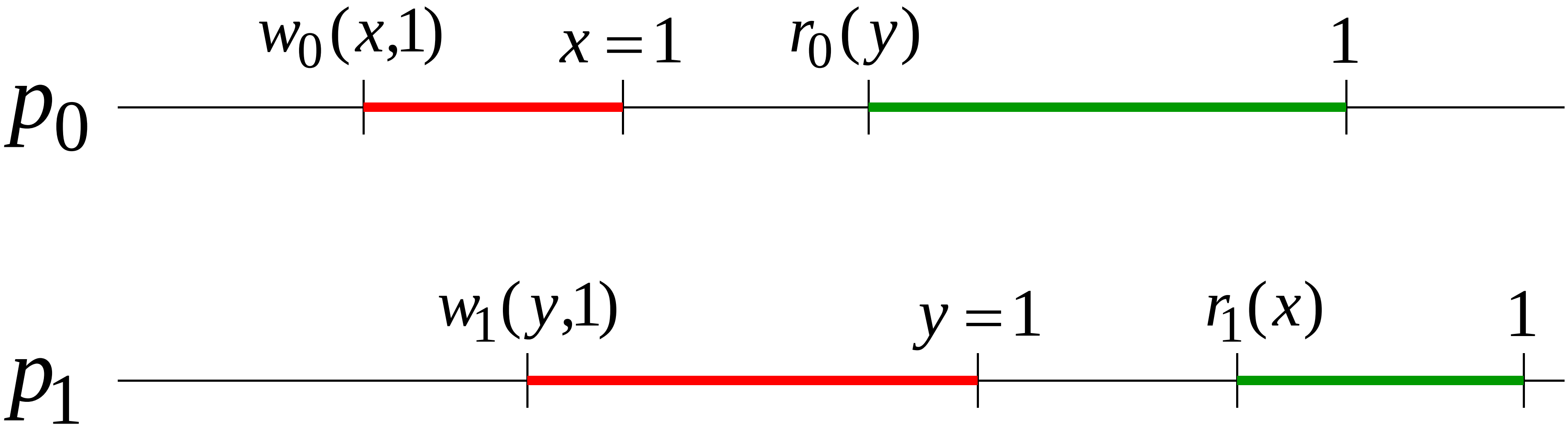
$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$



Nisu validna: $w_0\, r_0\, w_1\, r_1$ $\qquad$ $r_1\, w_0\, w_1\, r_0$

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

Sledeći primer

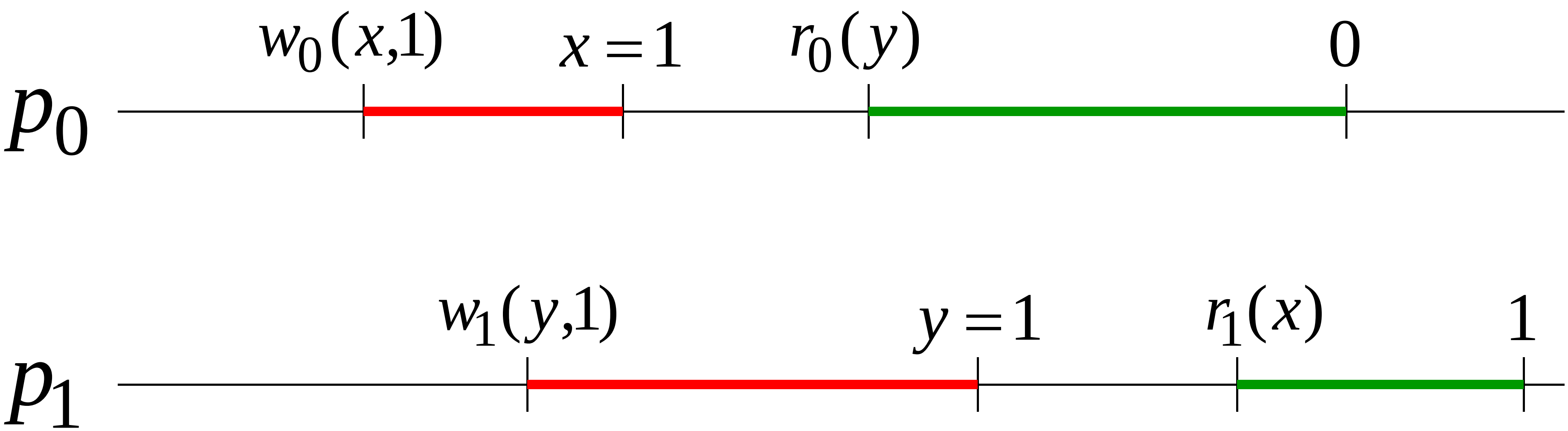


Validna: $w_0\, w_1\, r_0\, r_1$ $\quad$ $w_1\, w_0\, r_0\, r_1$ $\quad$ $w_0\, w_1\, r_1\, r_0$ $\quad$ $w_1\, w_0\, r_1\, r_0$

Nisu validna: $w_0\, r_0\, w_1\, r_1$ $\quad$ $r_1\, w_0\, w_1\, r_0$

Sledeći primer

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$



Validna: $w_0\, r_0\, w_1\, r_1$ $\quad r_0\, w_0\, w_1\, r_1$

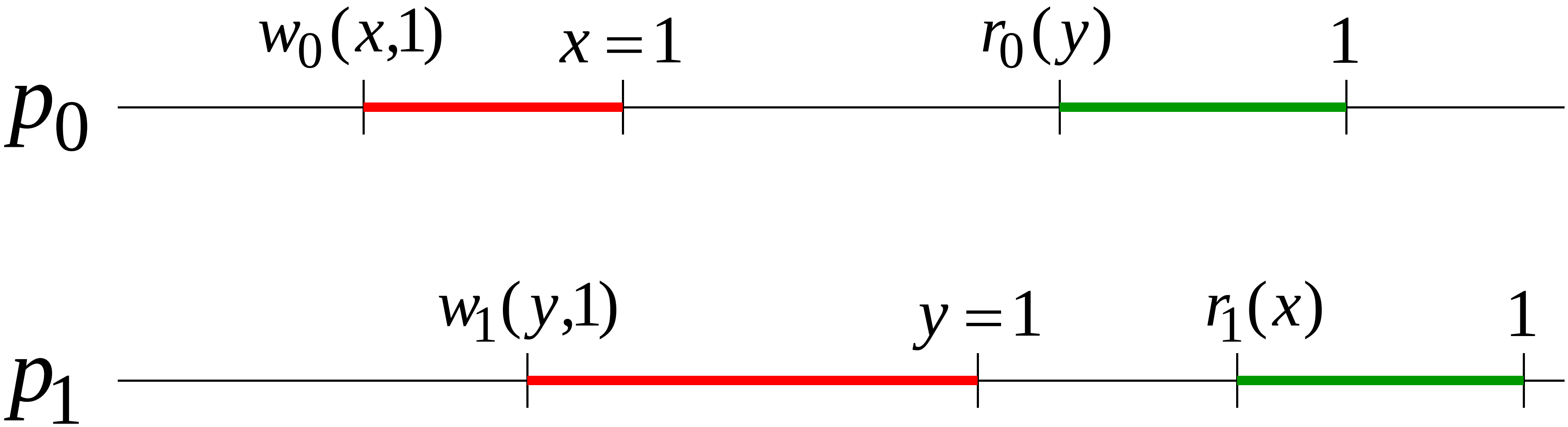
Nisu validna: $w_0\, w_1\, r_0\, r_1$ $\quad w_1\, w_0\, r_0\, r_1$

$w_0\, w_1\, r_1\, r_0$ $\quad w_1\, w_0\, r_1\, r_0$

Mogućnost linearizacije
(Linearizability)

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

Može se linearizovati



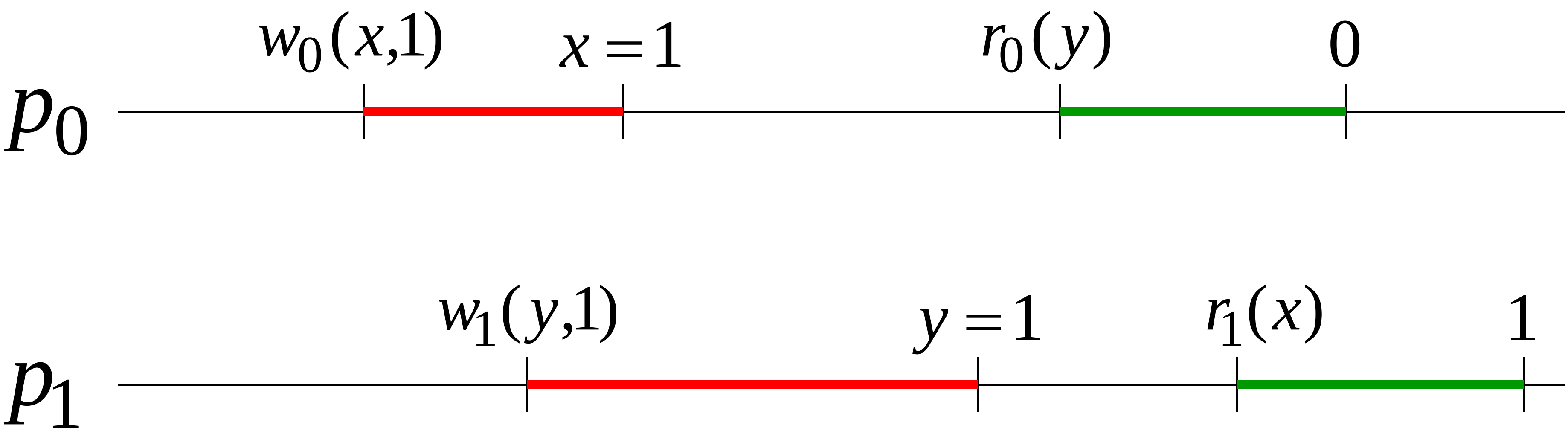
Postoji validan sekvencijalni raspored takav da:

događaji idu u redosledu realnog vremena u kom su izvršeni

$w_0\ w_1\ r_0\ r_1 \qquad w_1\ w_0\ r_0\ r_1 \qquad w_0\ w_1\ r_1\ r_0 \qquad w_1\ w_0\ r_1\ r_0$

Ne može se linearizovati

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

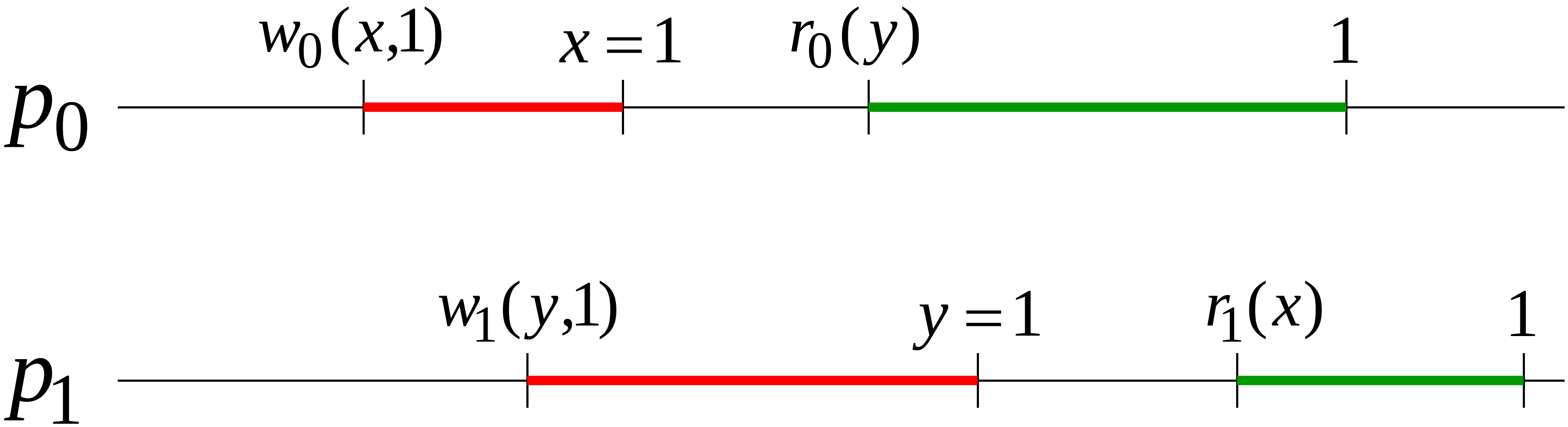


$$r_0 < w_1 \qquad w_1 < r_0 \qquad \Rightarrow \qquad r_0 < r_0$$

Nemoguće!

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

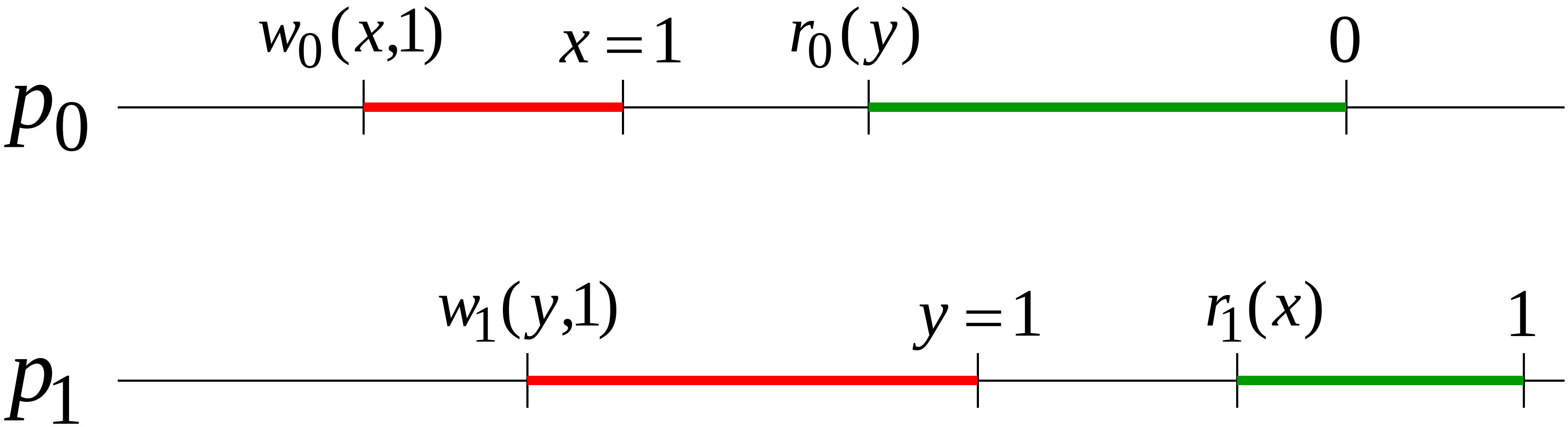
Može se linearizovati



$w_0\, w_1\, r_0\, r_1 \quad w_1\, w_0\, r_0\, r_1 \quad w_0\, w_1\, r_1\, r_0 \quad w_1\, w_0\, r_1\, r_0$

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

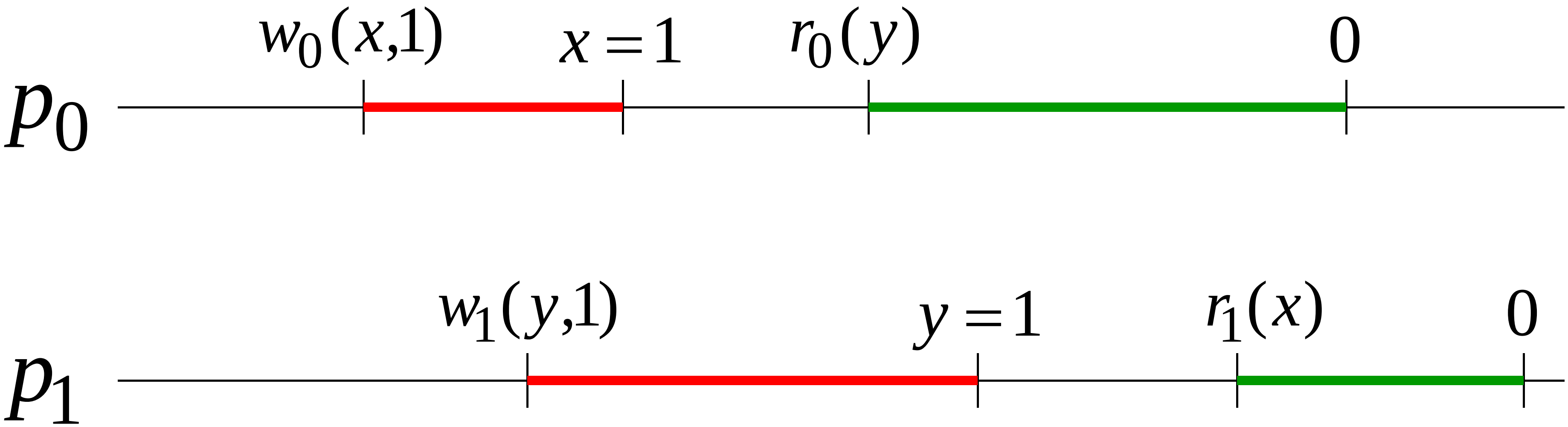
Može se linearizovati



$$w_0 \; r_0 \; w_1 \; r_1$$

$$\boxed{\begin{array}{l} x = 0 \\ y = 0 \end{array}}$$

Ne može se linearizovati

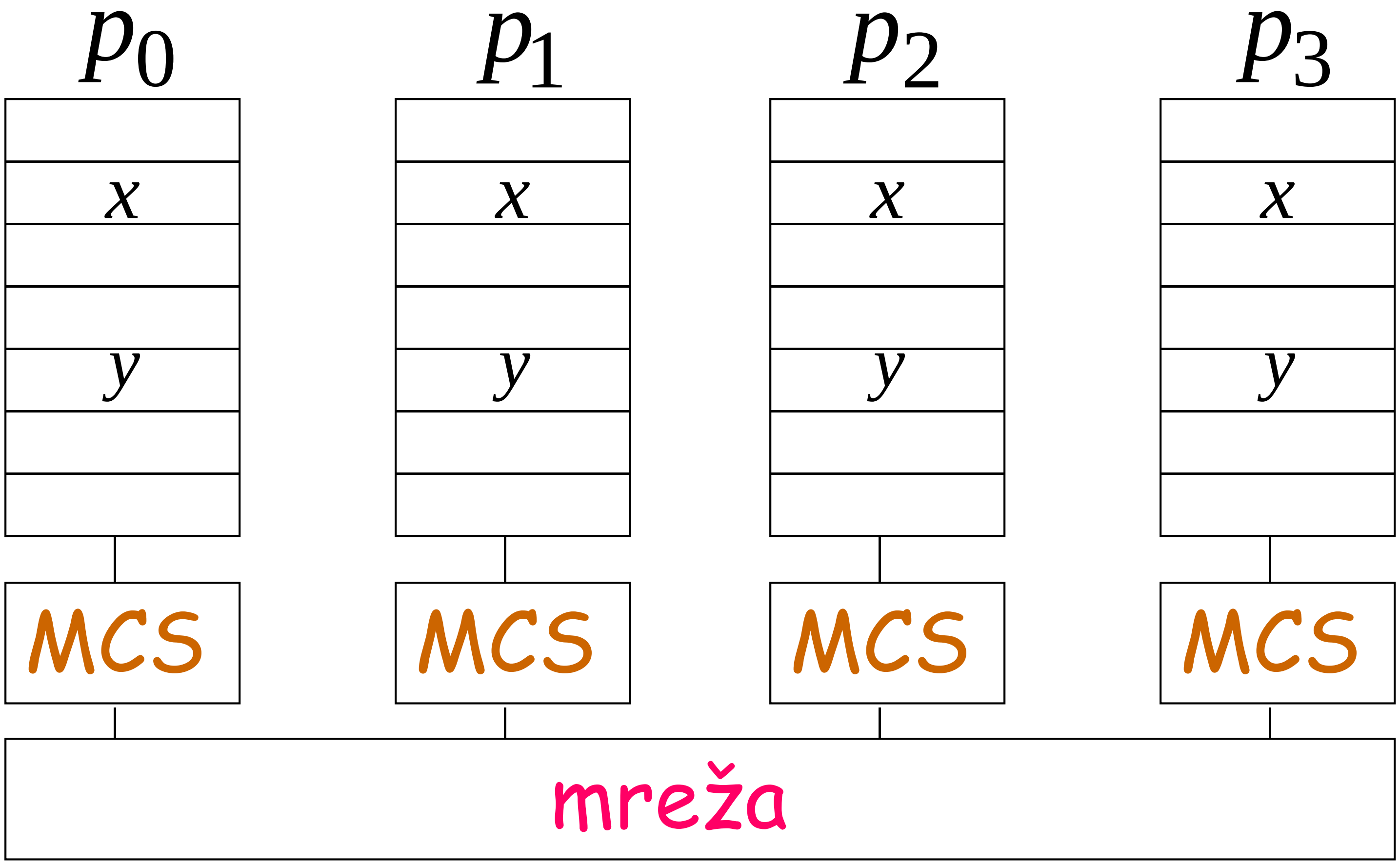


$$r_1 < w_0 \qquad w_0 < r_1 \qquad \Rightarrow \qquad r_1 < r_1$$

Nemoguće!

Želimo da implementiramo
MCS (Memory Consistency System)
koji se može linearizovati:

Svaki procesor posmatra događaje u
nekom sekvencijalnom redosledu koji
je rezultat linearizacije



MCS: omogućava linearizaciju

Algoritam za omogućavanje linearizacije

DOGAĐAJ READ za procesor p_i :

Kad se desi $r_i(x)$:

Totalno uređeno slanje svima $r_i(x)$

Nakon prijema poruke $r_j(x)$:

Ako $i == j$ vrati lokalnu vrednost x

Algoritam za omogućavanje linearizacije

DOGAĐAJ WRITE za procesor p_i :

Kad se desi $w_i(x,v)$:

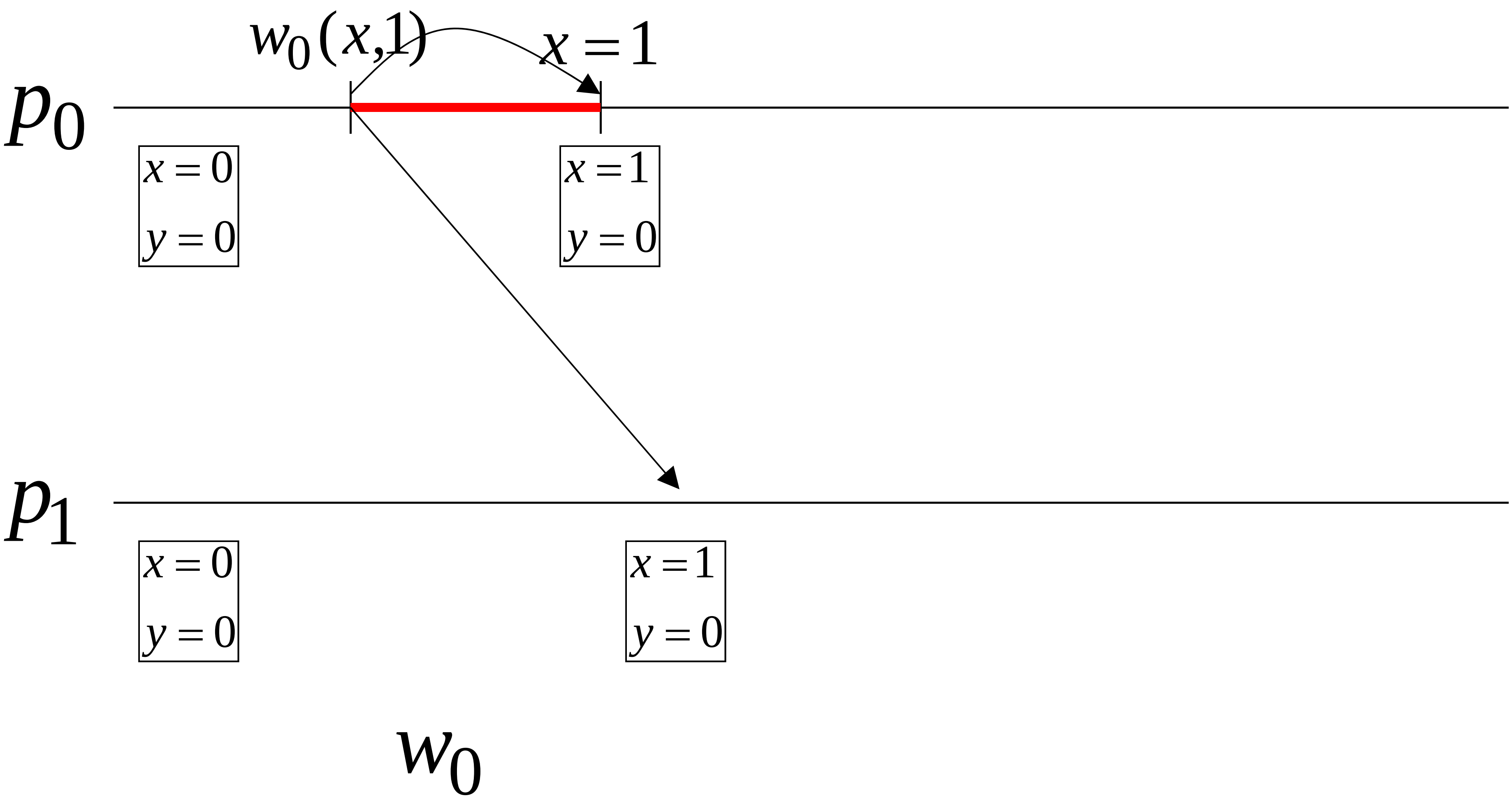
Totalno uređeno slanje svima $w_i(x,v)$

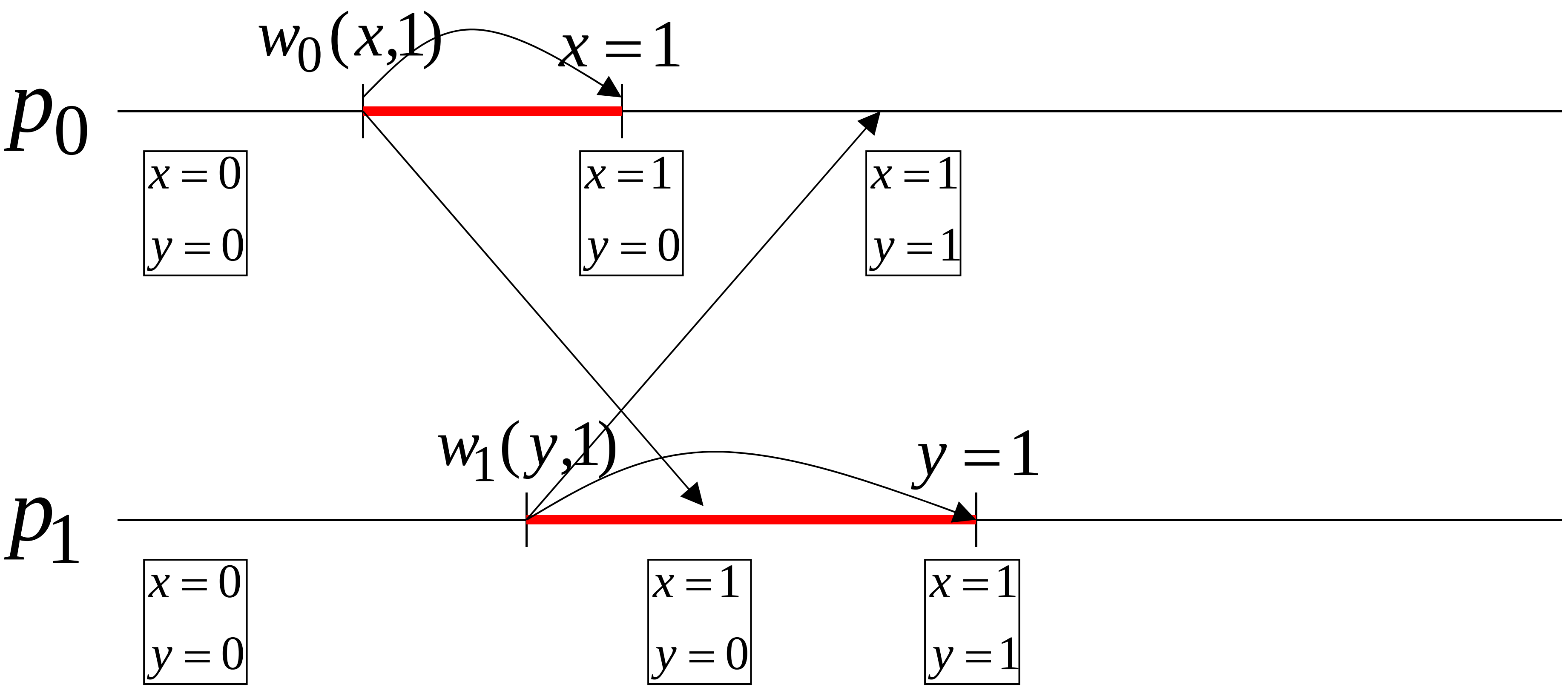
Nakon prijema poruke $w_j(x,v)$:

Postavi lokalnu vrednost $x = v$

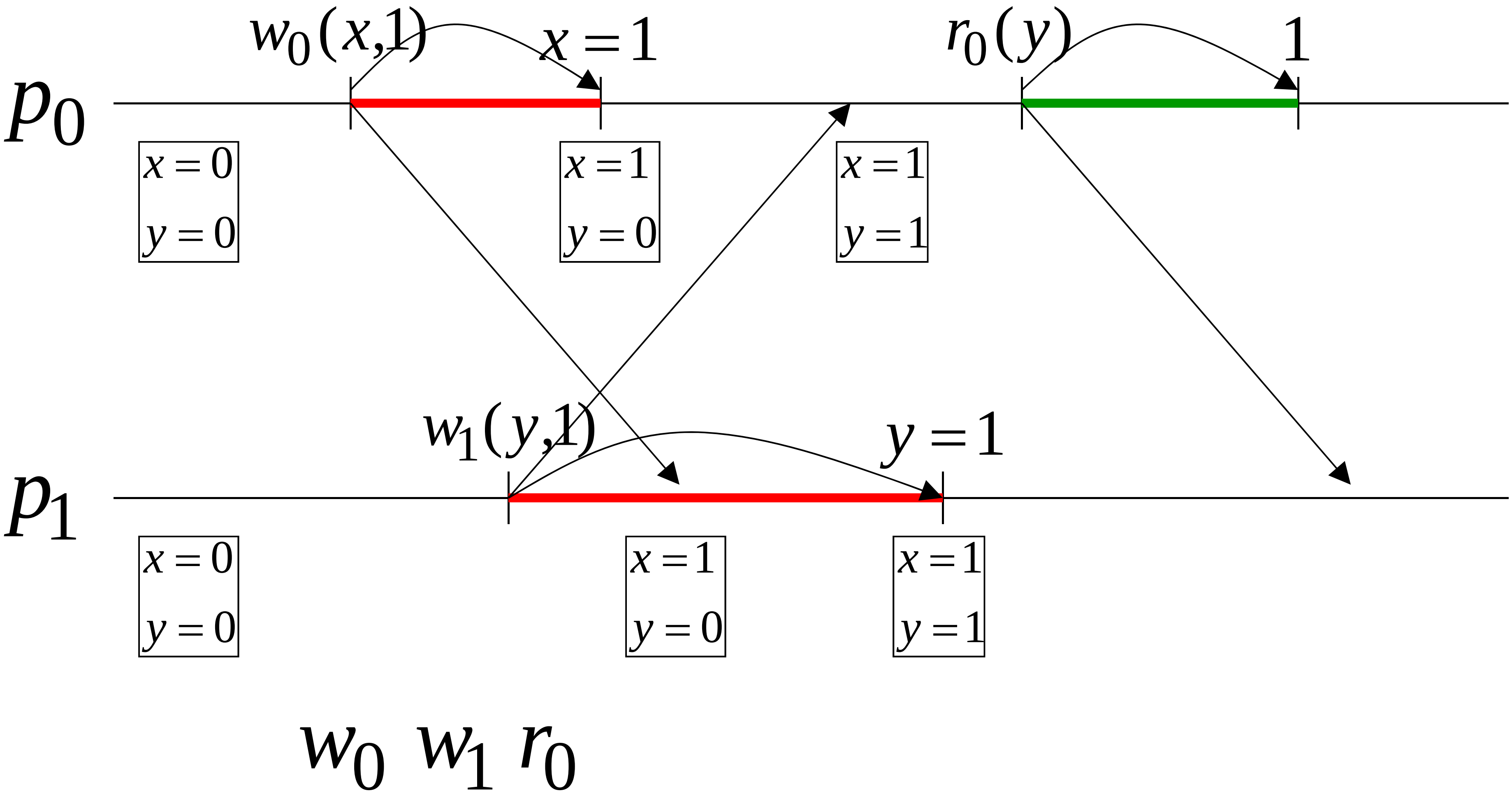
(ako je $i == j$ takođe potvrdi)

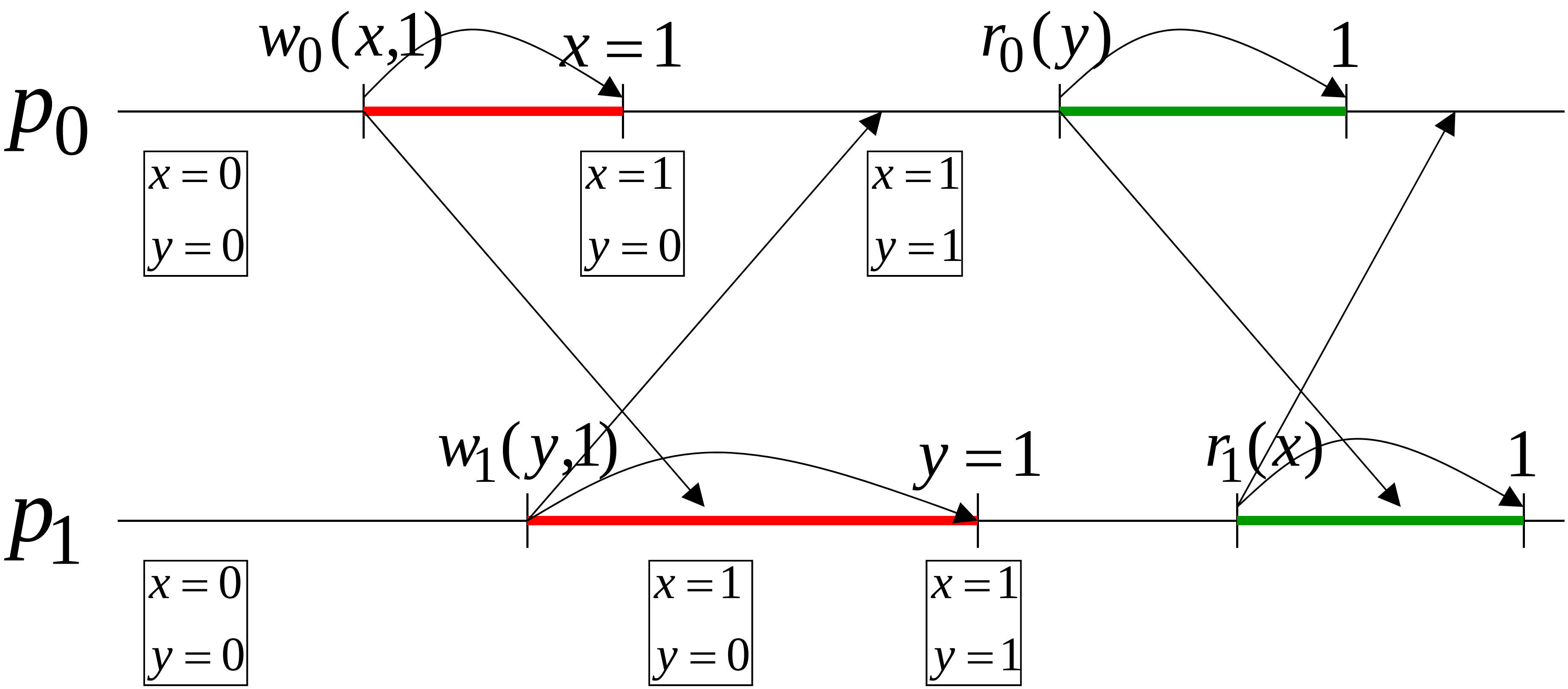
Primer izvršenja





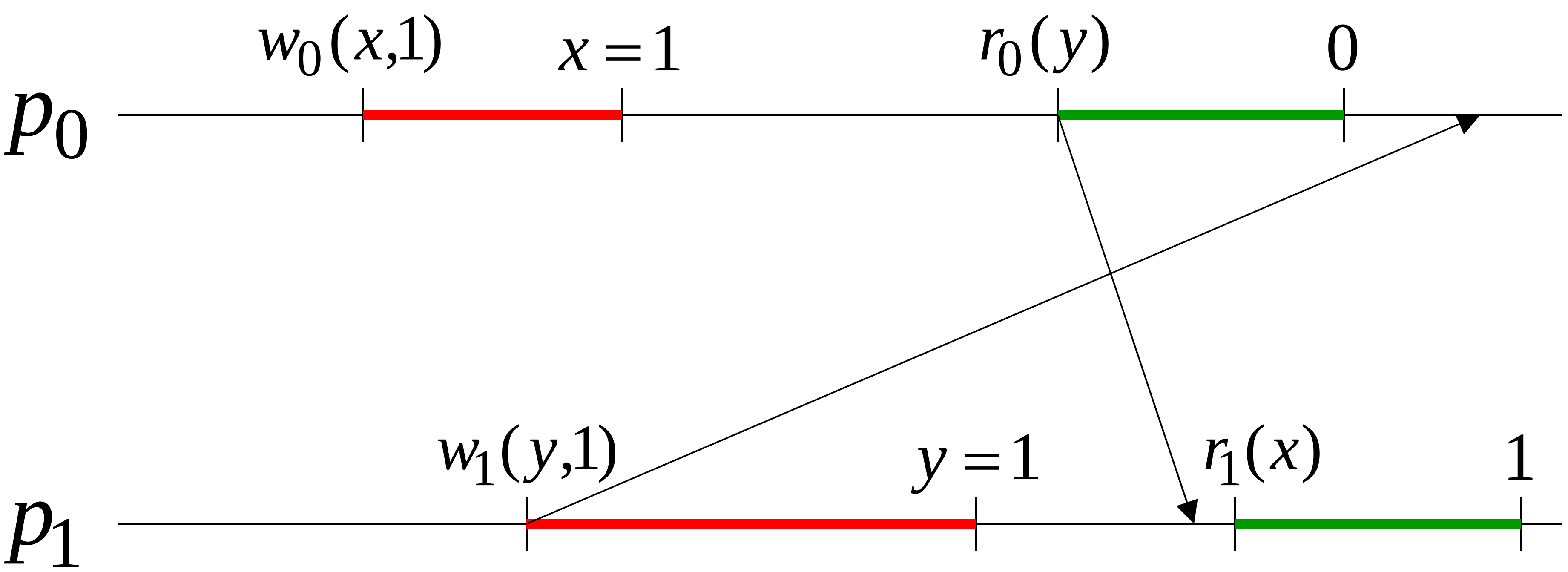
$w_0 \; w_1$


$w_0(x,1)$
$x=1$
$r_0(y)$
1
p_0
$x=0$
$y=0$
$x=1$
$y=0$
$x=1$
$y=1$
$w_1(y,1)$
$y=1$
p_1
$x=0$
$y=0$
$x=1$
$y=0$
$x=1$
$y=1$
w_0 w_1 r_0



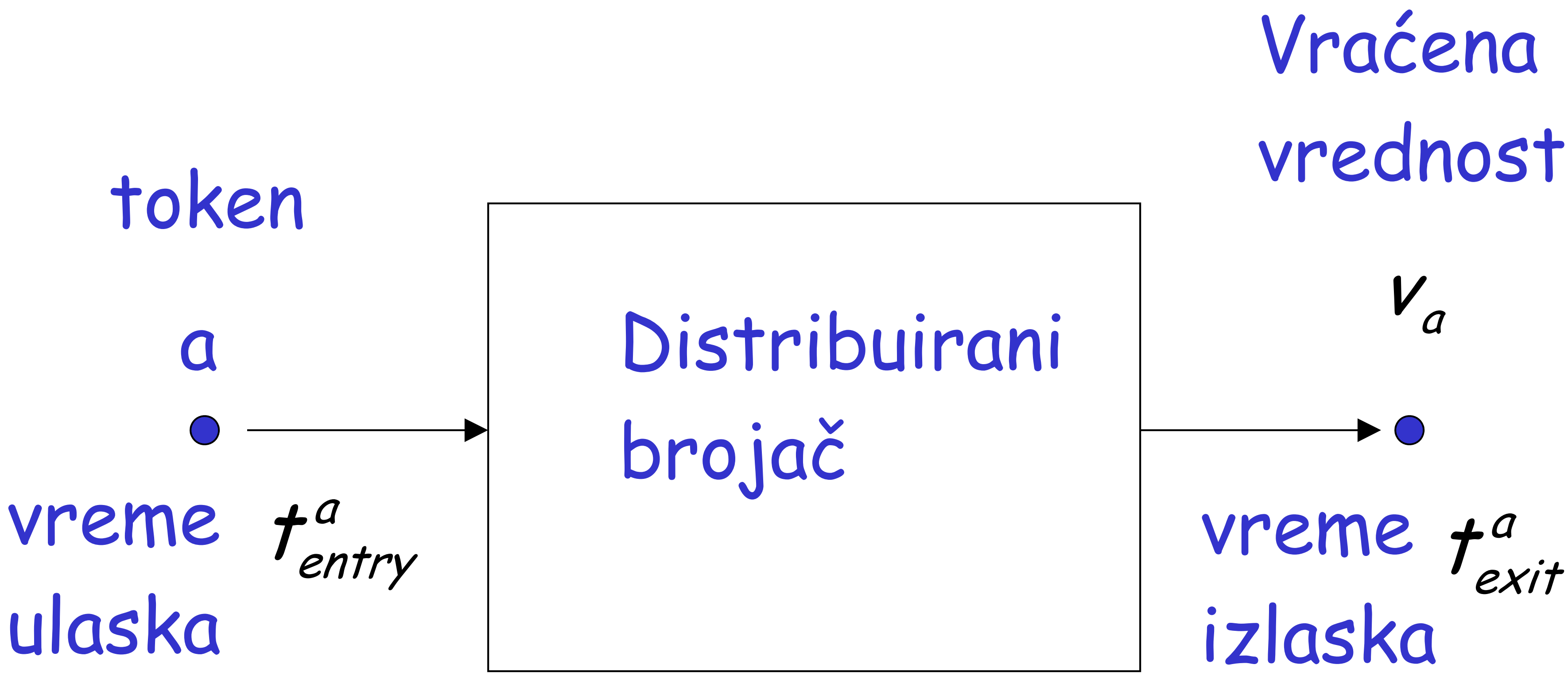
$$w_0 \; w_1 \; r_0 \; r_1$$

Nemoguće izvršenje



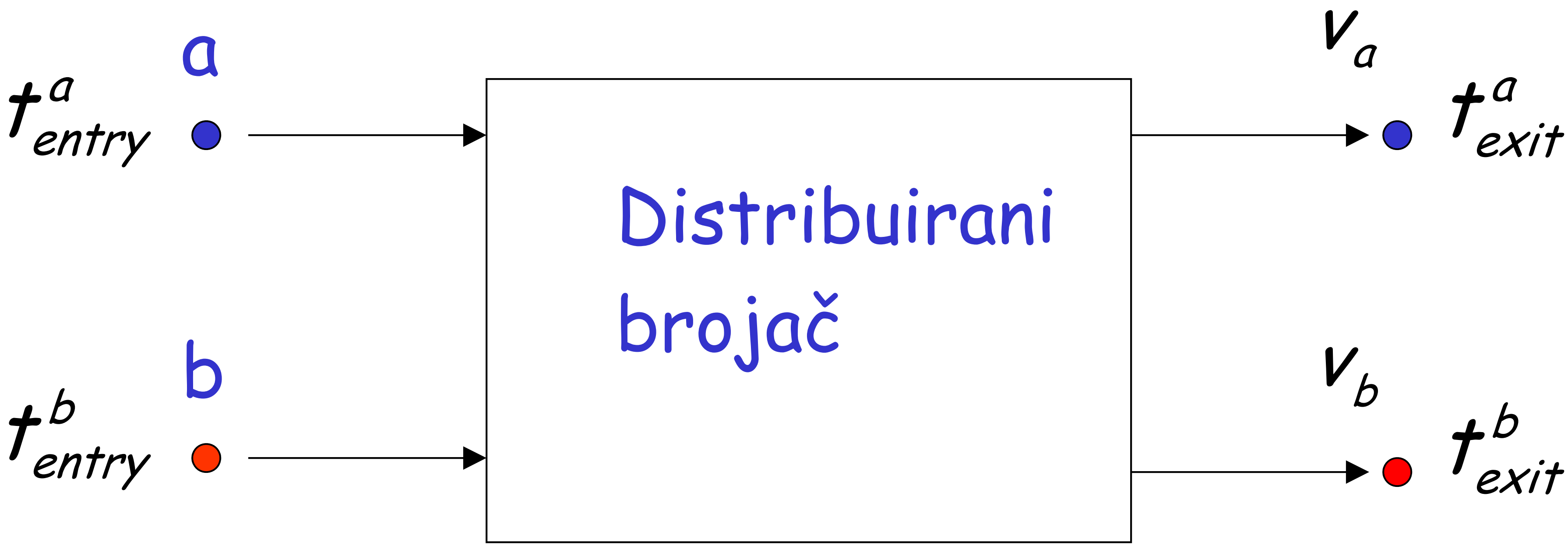
(izvršenje koje se ne može linearizovati)

Mreže za brojanje čije izvršenje se može linearizovati



Vreme prolaska tokena: $t^a_{exit} - t^a_{entry}$

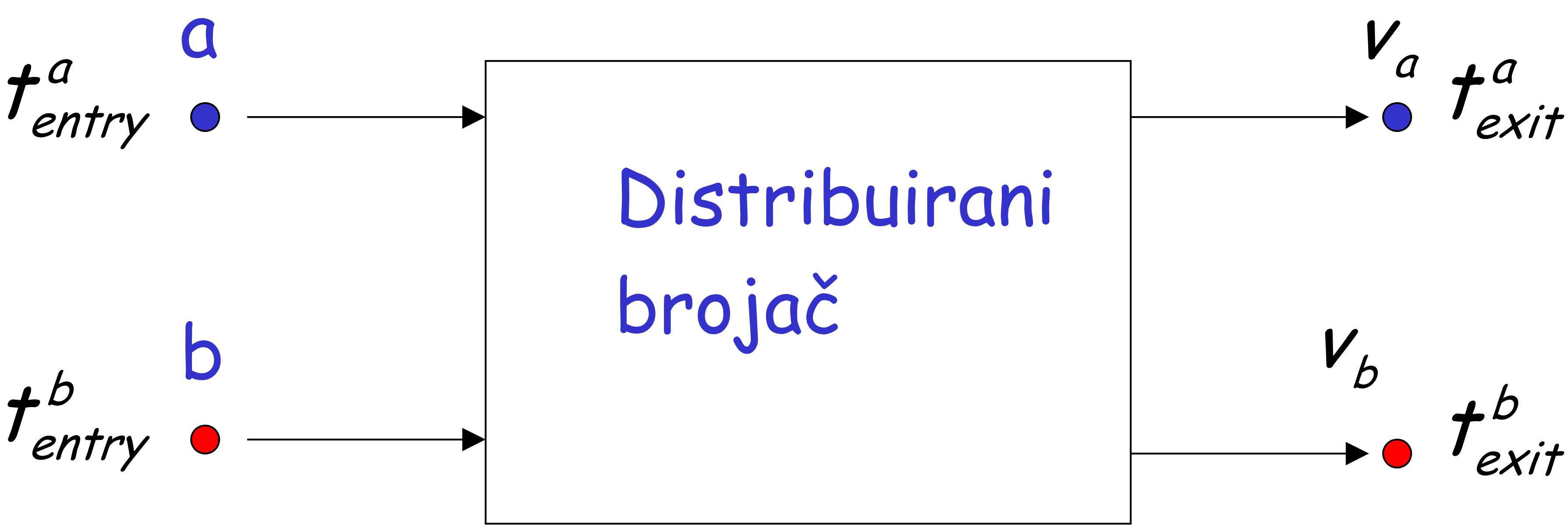
(različito za različite tokene)



Vremena ulaska su proizvoljna,
pošto tokeni nailaze u proizvoljnim trenutcima

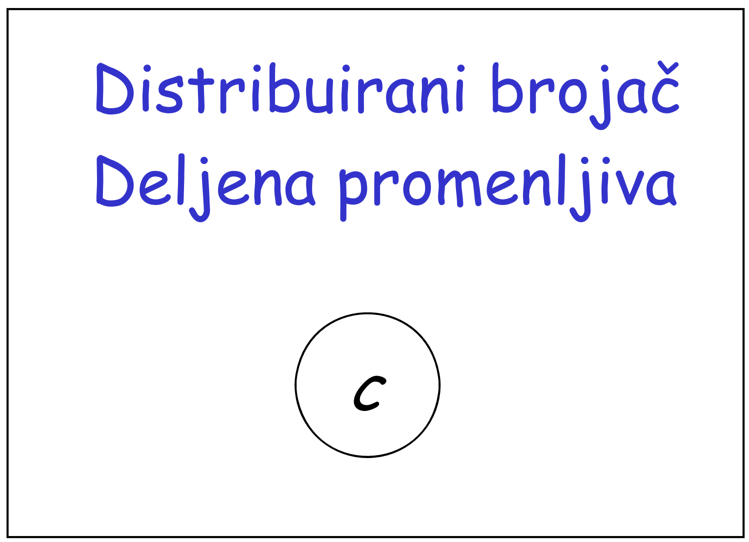
U izvršenjima koja se mogu linearizovati:

Ako $t^a_{exit} < t^b_{entry}$ onda $v_a < v_b$

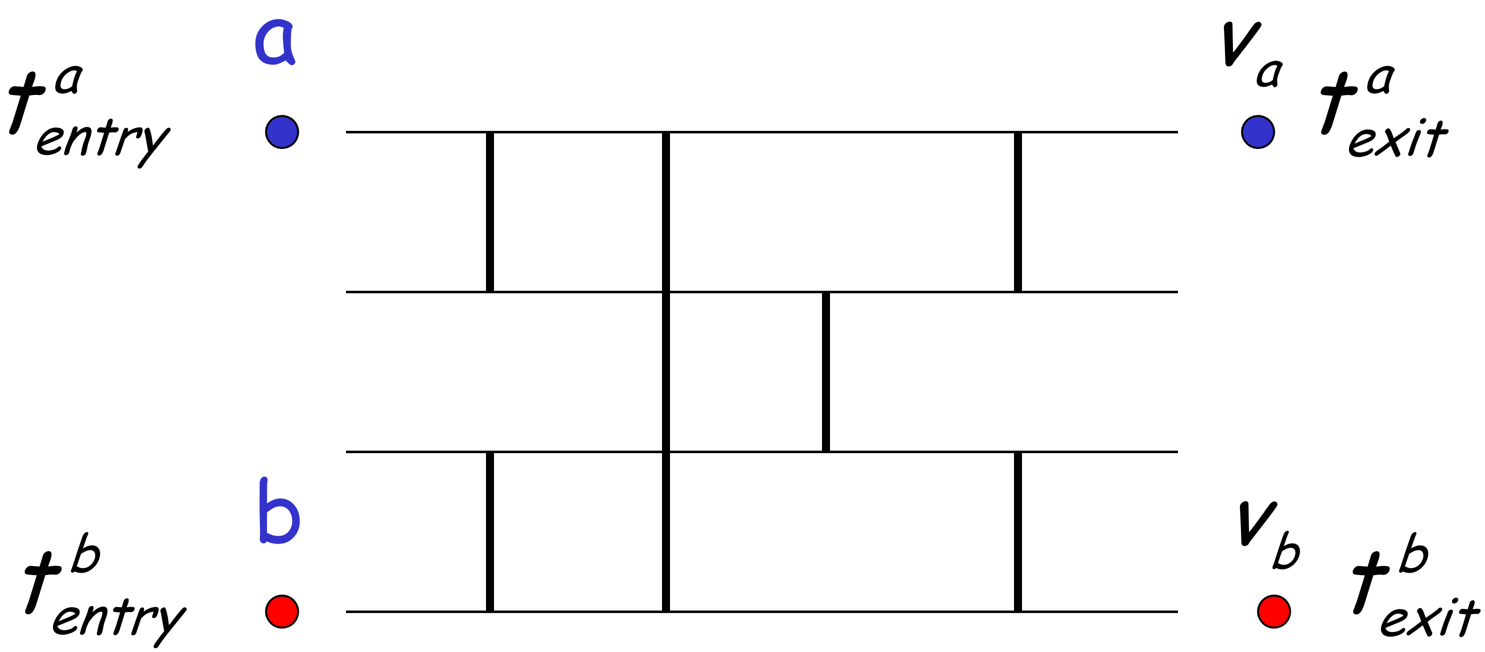


Naime: novi tokeni uzimaju veće vrednosti

Brojač sa jednom deljenom promenljivom se može linearizovati

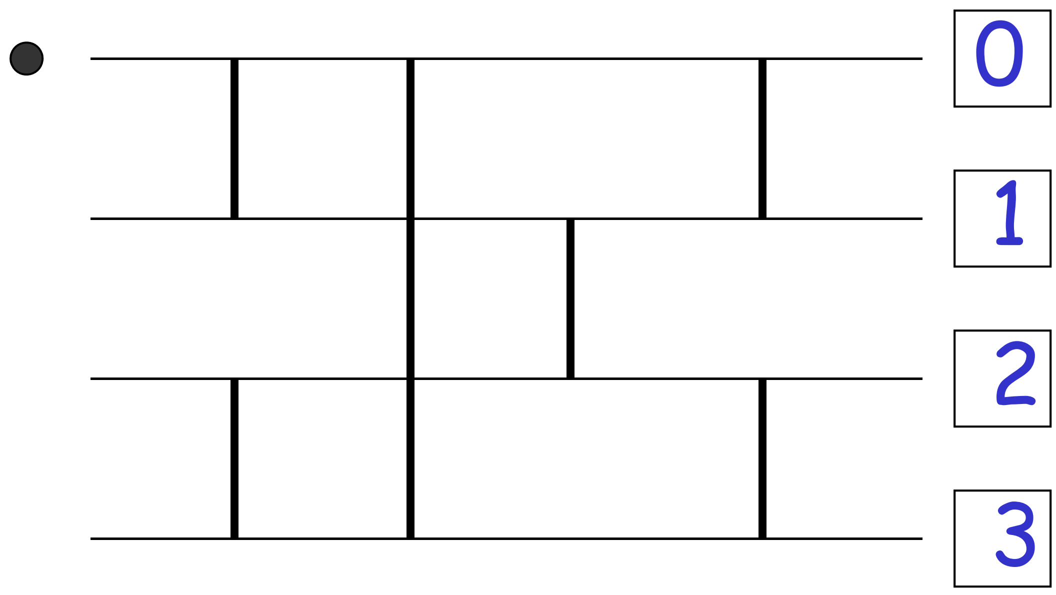


Proste mreže za broja. se <u>ne mogu</u> linearizovati

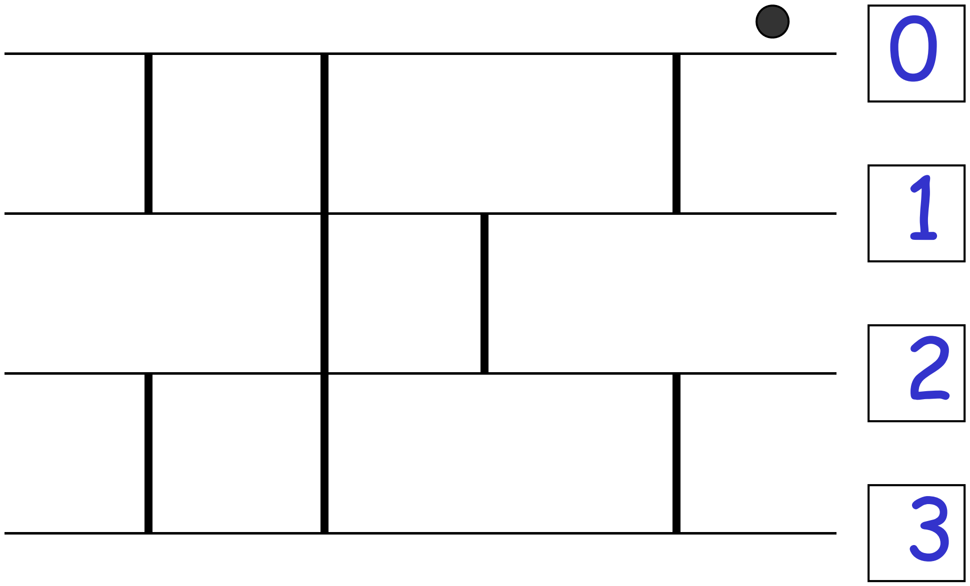


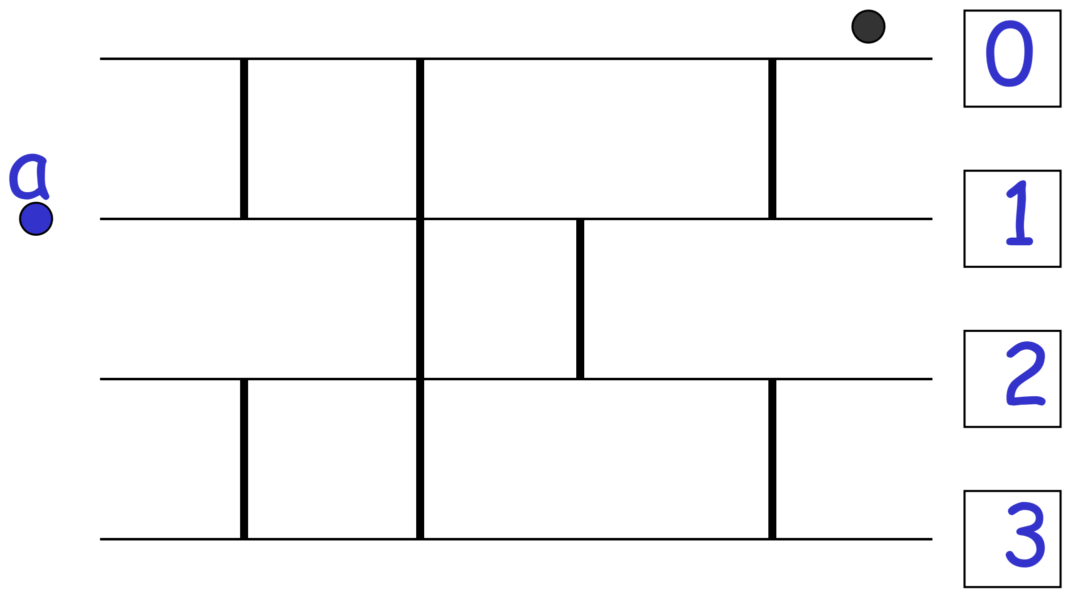
Postoje izvršenja takva da je:

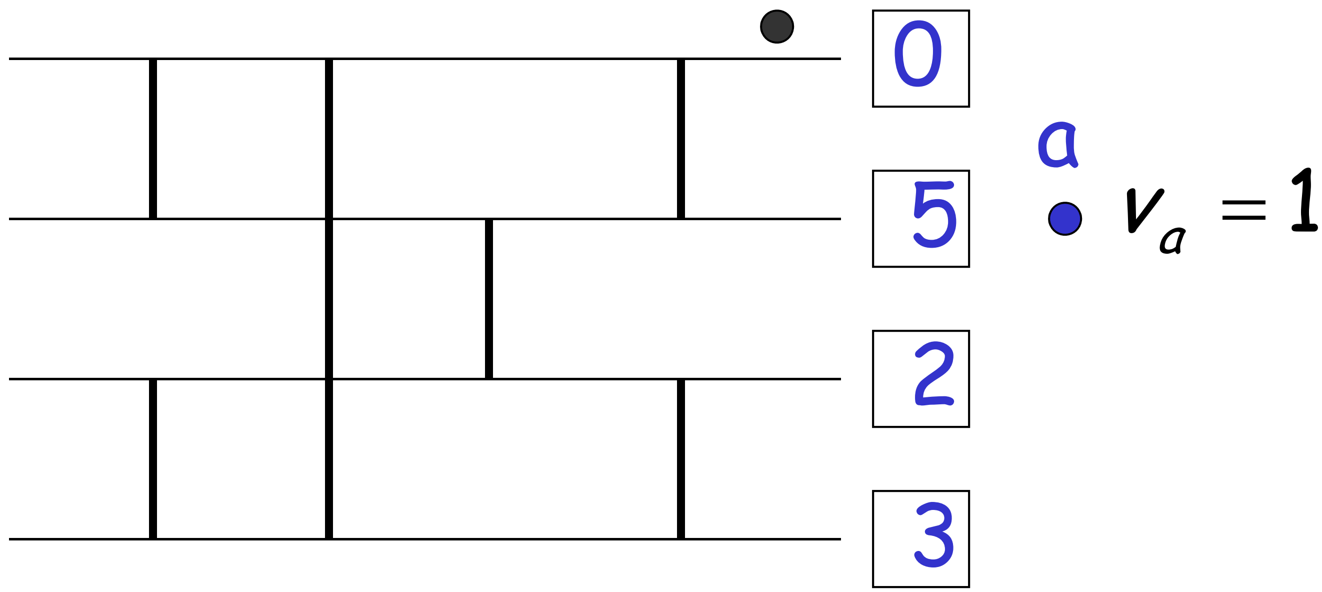
$$t^a_{exit} < t^b_{entry} \quad \text{i} \quad v_a > v_b$$


0
1
2
3

token se ne
pomera na izlaz

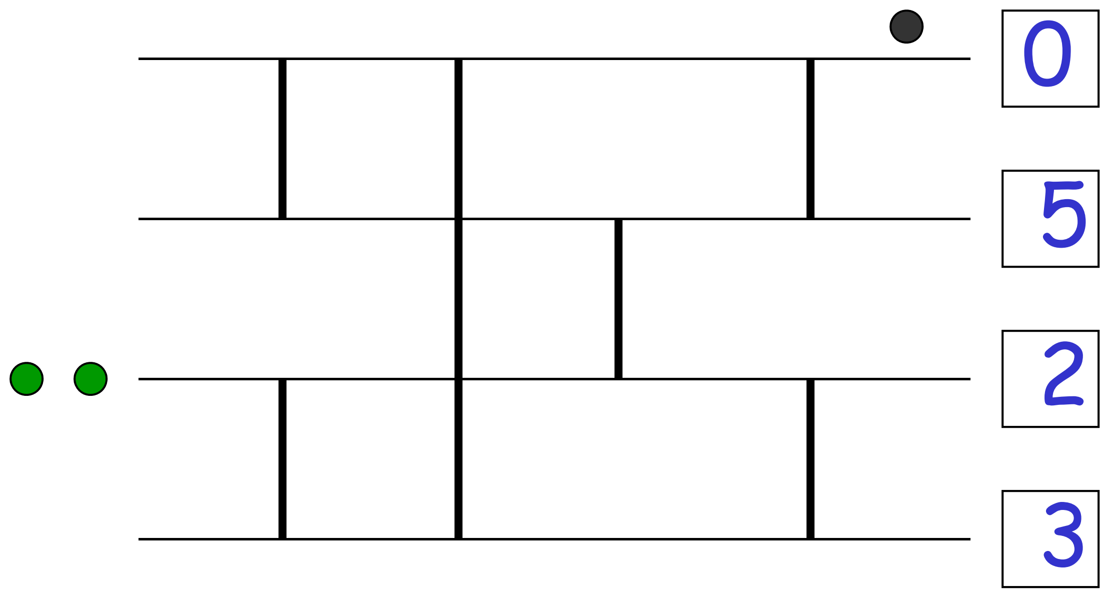



a
0
1
2
3


0
5
2
3
a
$v_a = 1$

a

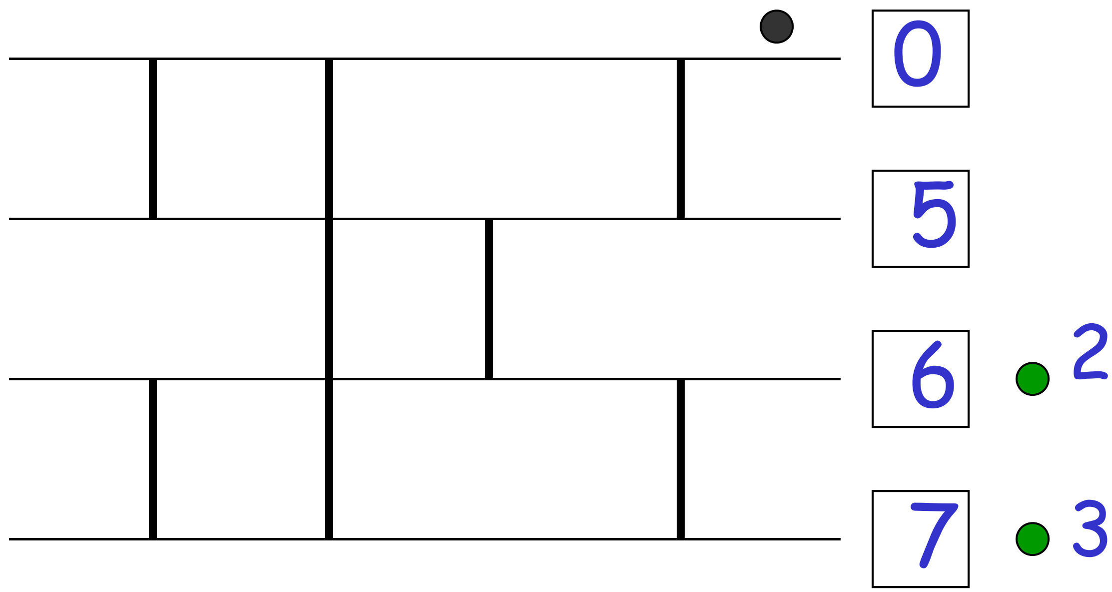
$v_a = 1$



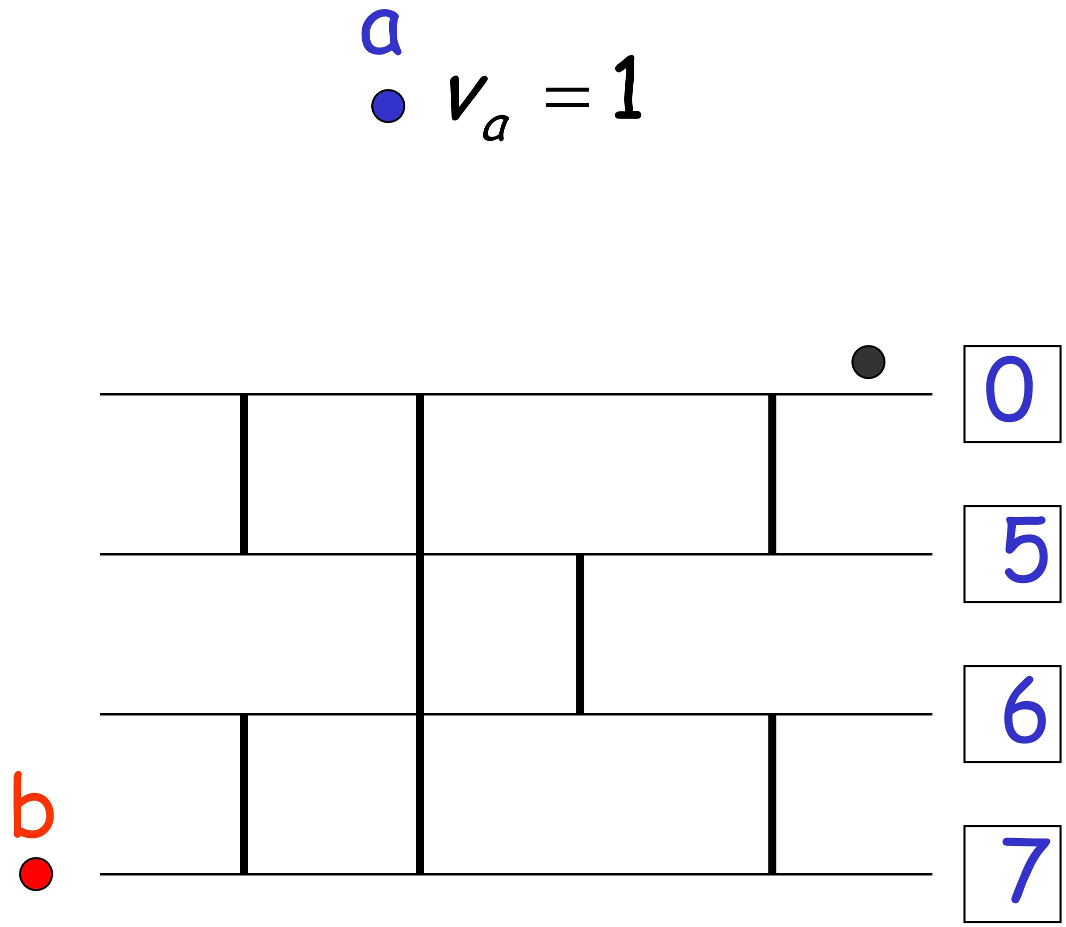
Još dva tokena nailaze

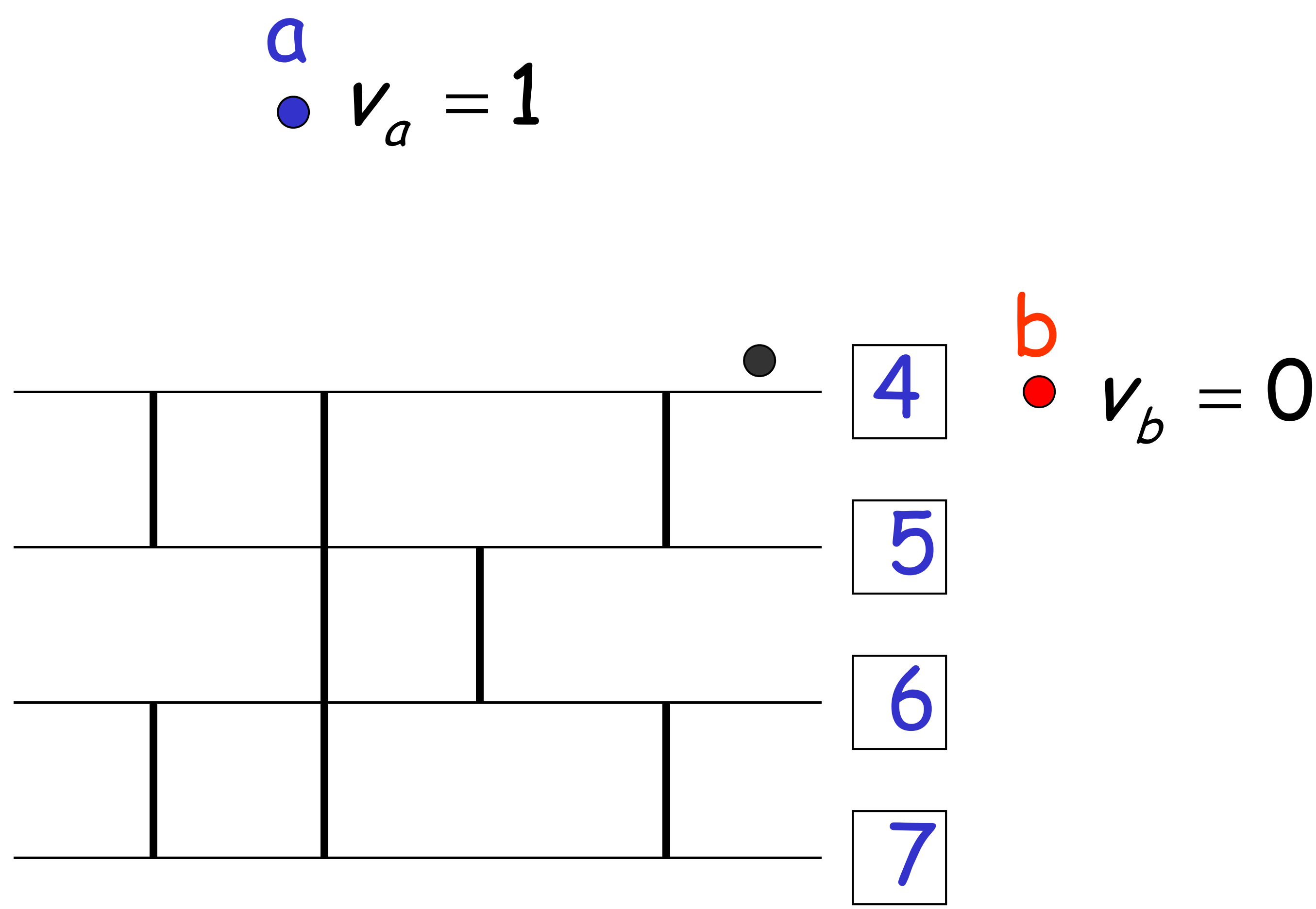
a

$v_a = 1$



Još dva tokena nailaze


a
$v_a = 1$
0
5
6
7
b



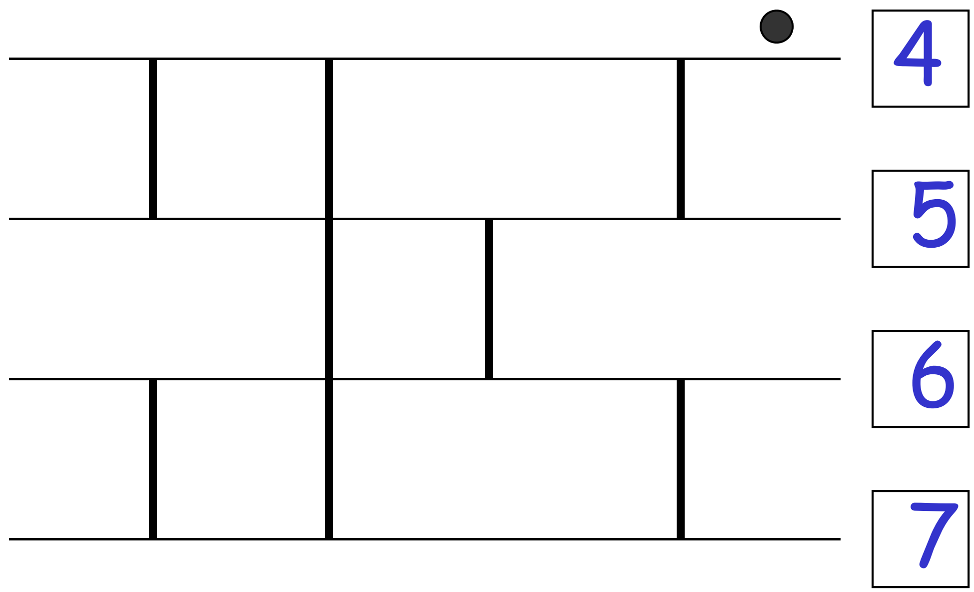
a
$v_a = 1$
b
$v_b = 0$
4
5
6
7

a

$v_a = 1$

b

$v_b = 0$



$t^a_{exit} < t^b_{entry}$ i $v_a > v_b$

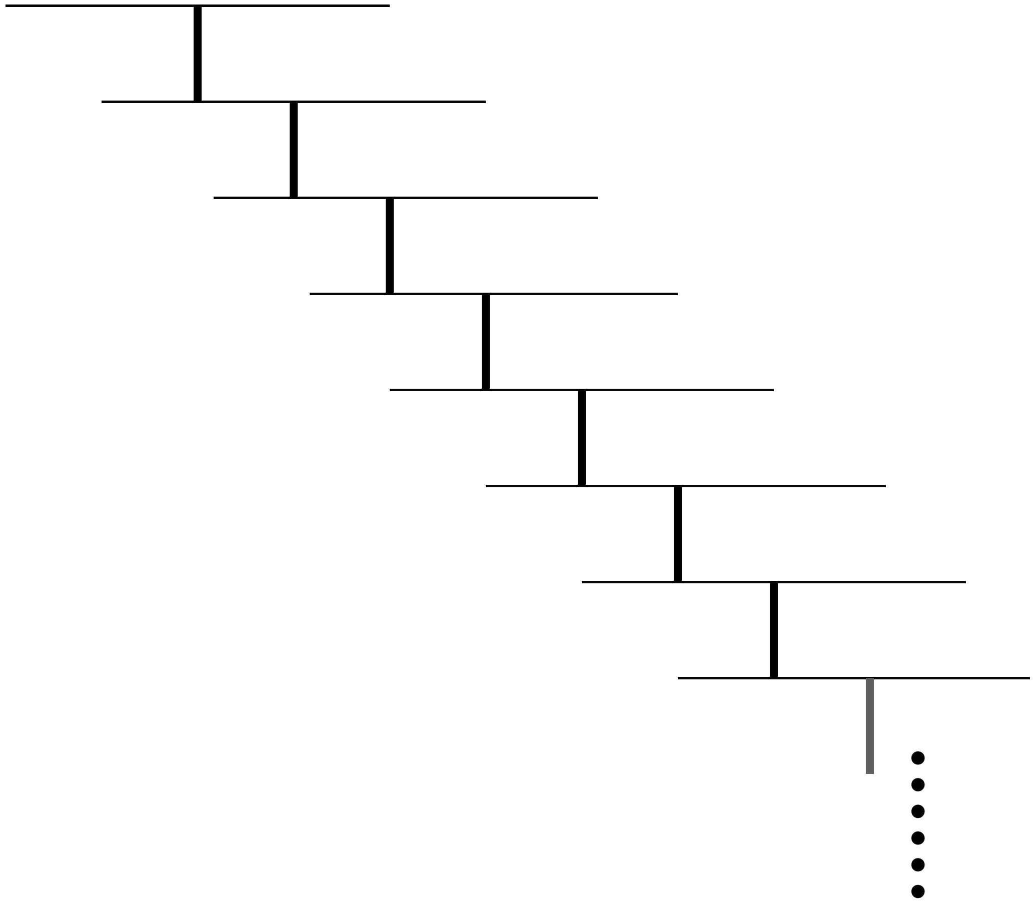
Narušavanje mogućnosti linearizacije!

Mreže za brojanje koje se mogu linearizovati

Predpost. da ima n procesa
koji mogu da izdaju tokene

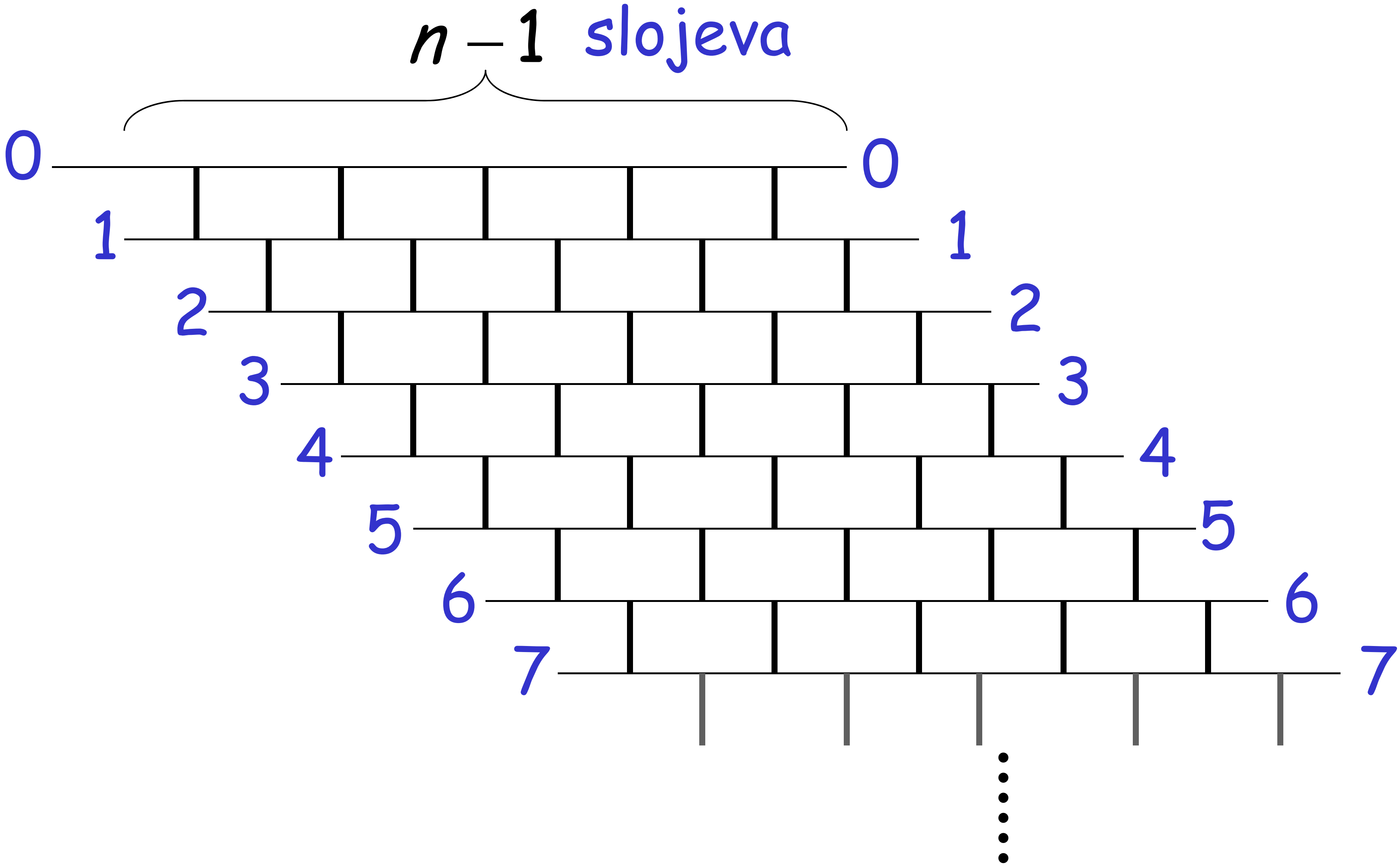
U bilo kom trenutku vremena
ima najviše n tokena
u mreži za brojanje

Jedan sloj



Beskonačna dužina

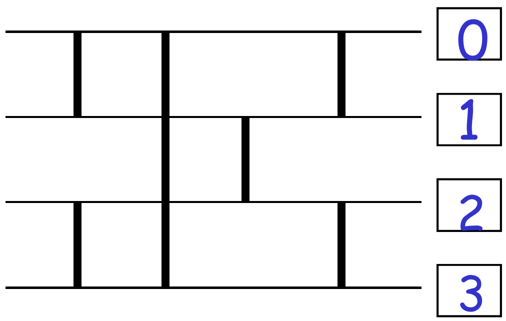
Kosa mreža

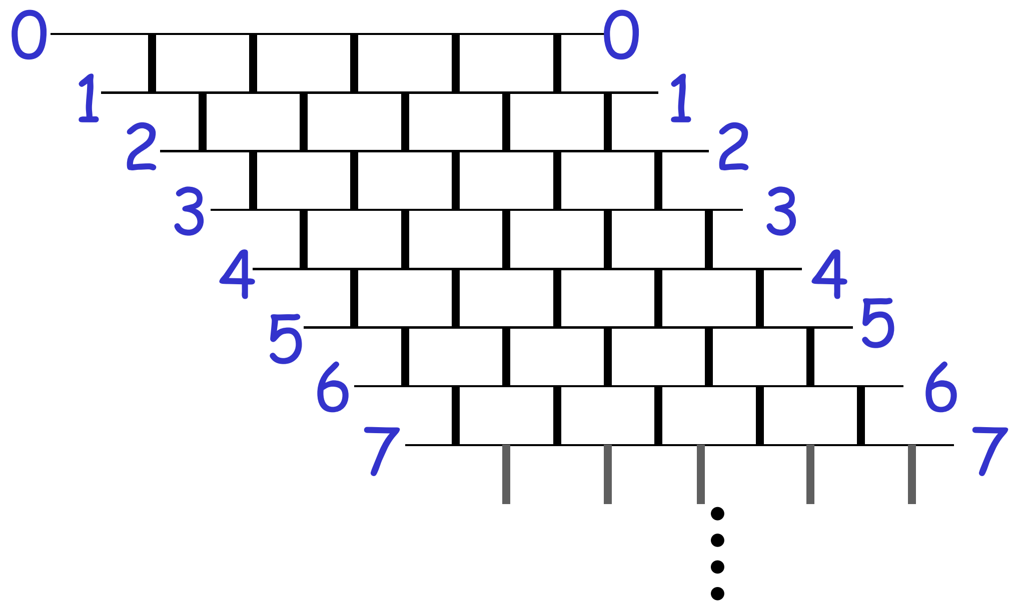


Mreža za brojanje koja se može linearizovati

regularna

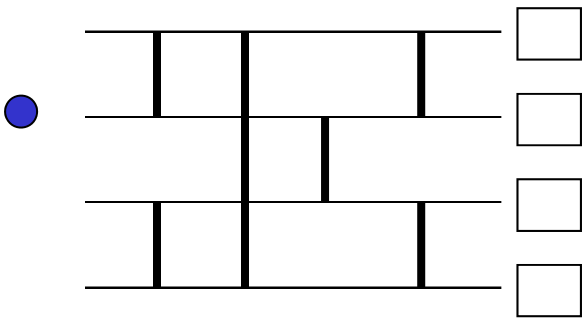
mreža za brojanje + kosa mreža



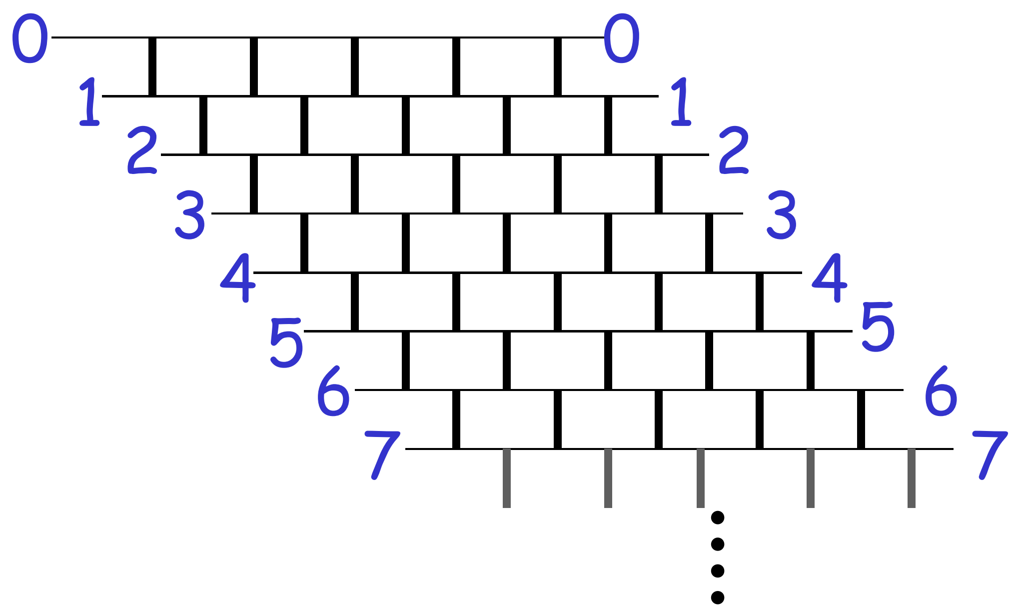


Token prvo ulazi u mrežu za brojanje a zatim u kosu mrežu

Mreža za brojanje

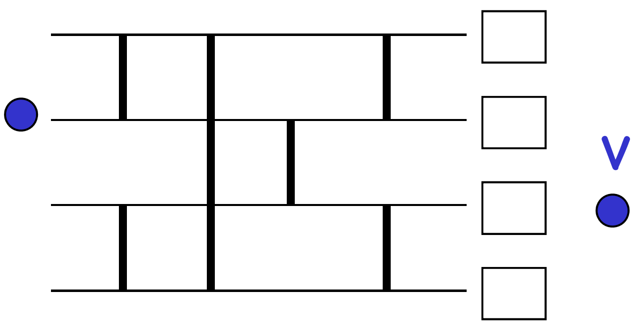


Kosa mreža

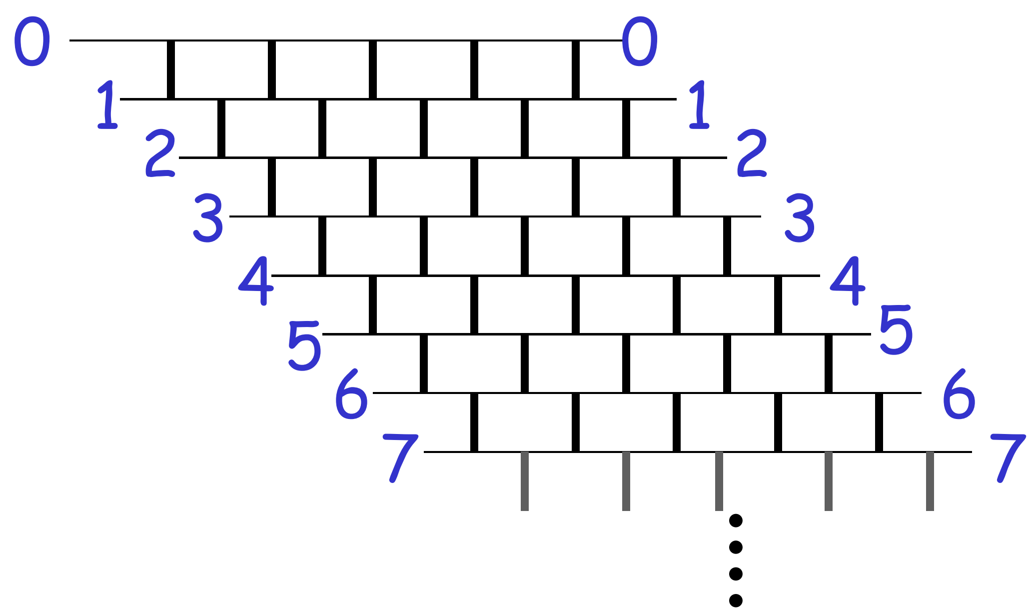


Token prvo dobija vrednost od mreže za brojanje

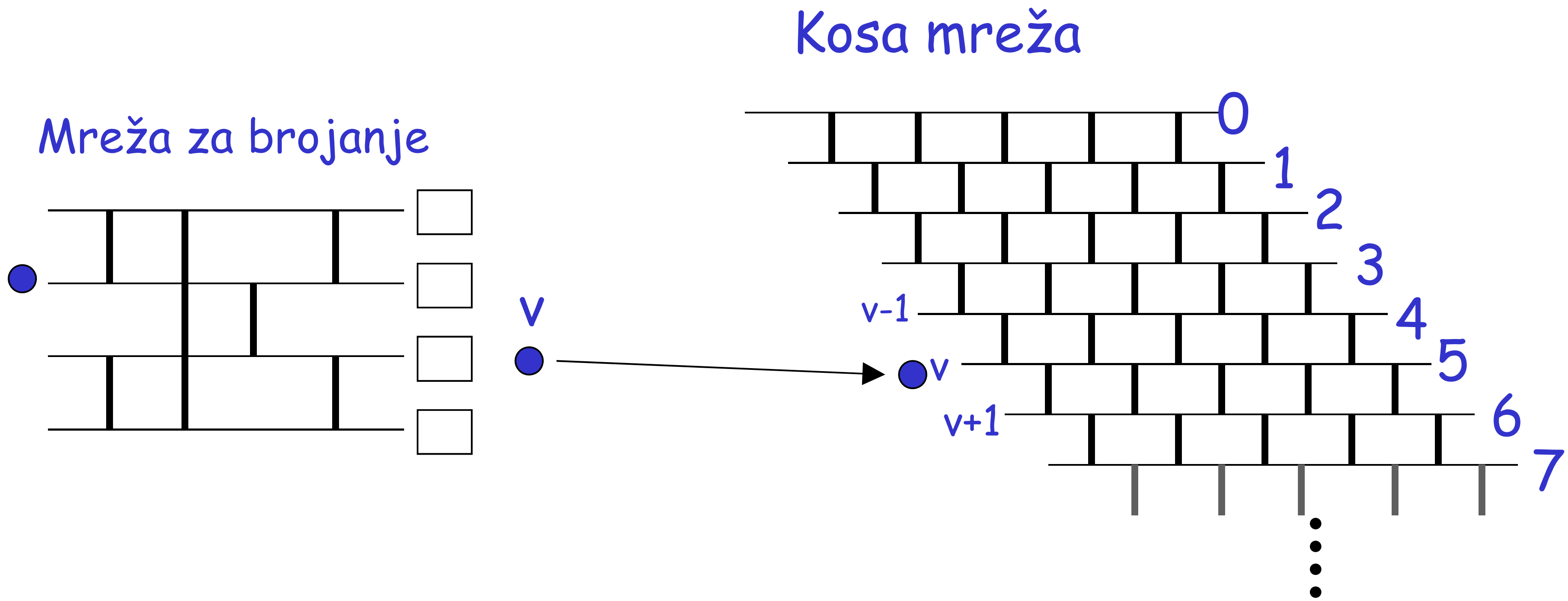
Mreža za brojanje



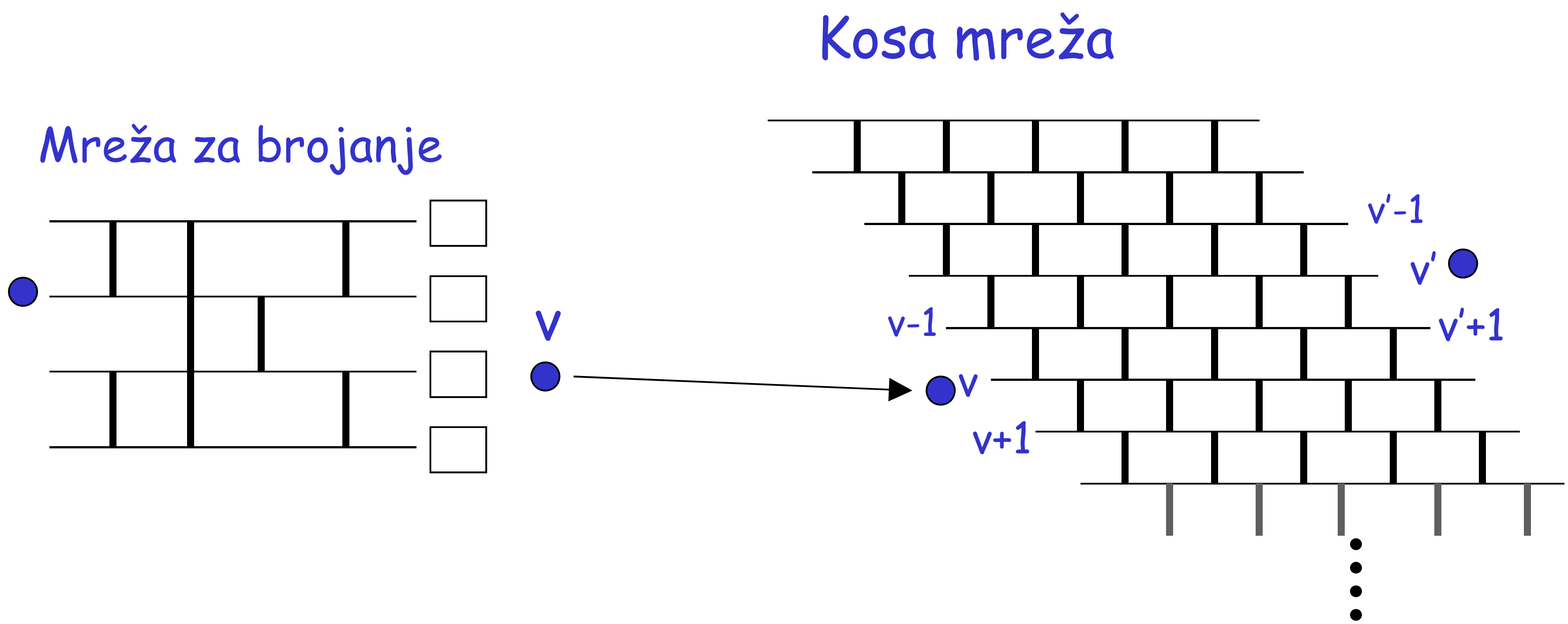
Kosa mreža



Token ulazi u kosu mrežu
Ulazni indeks je vrednost v

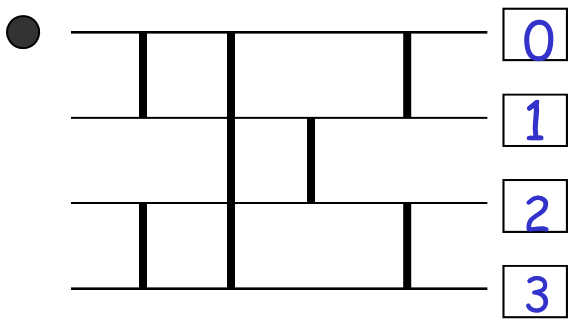


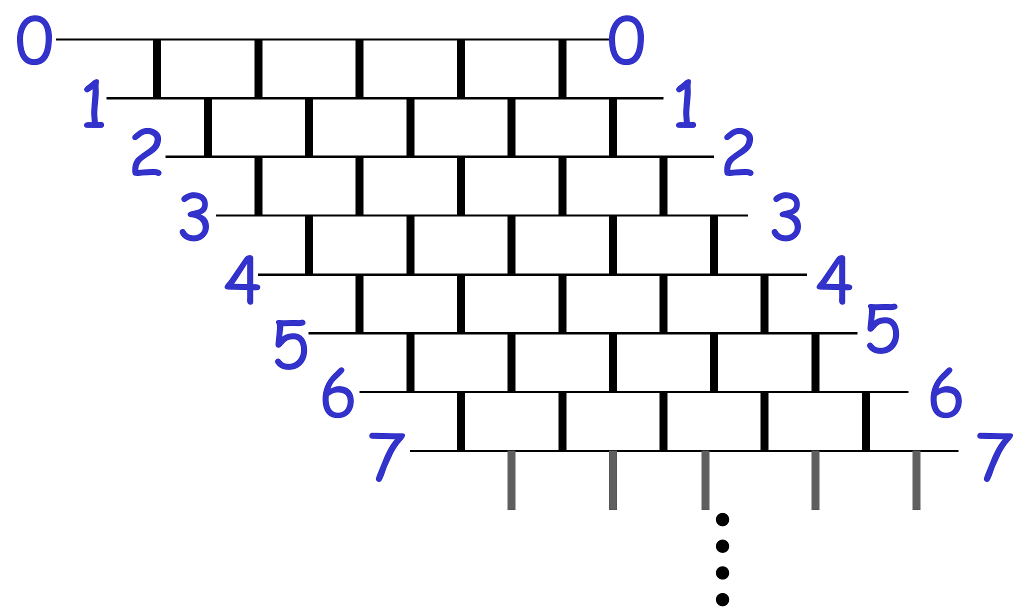
Token prolazi kroz kosu mrežu i rezultantna vrednost je izlazni indeks



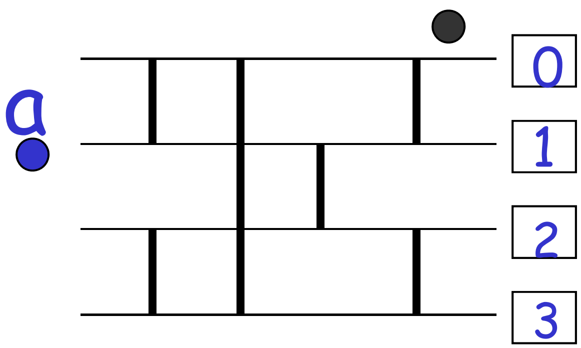
Rezultantna vrednost tokena: v'

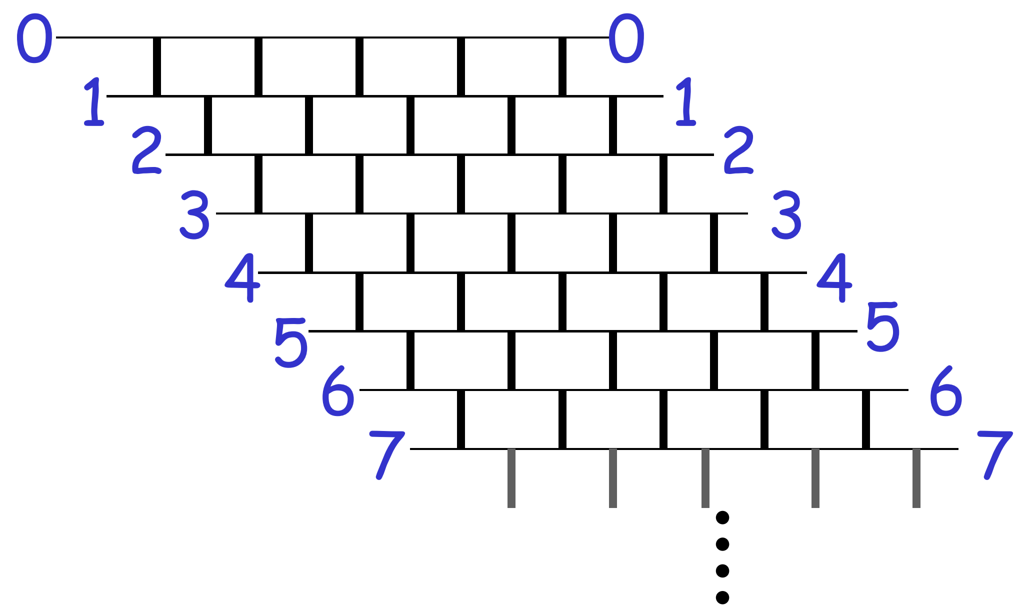
Primer izvršenja

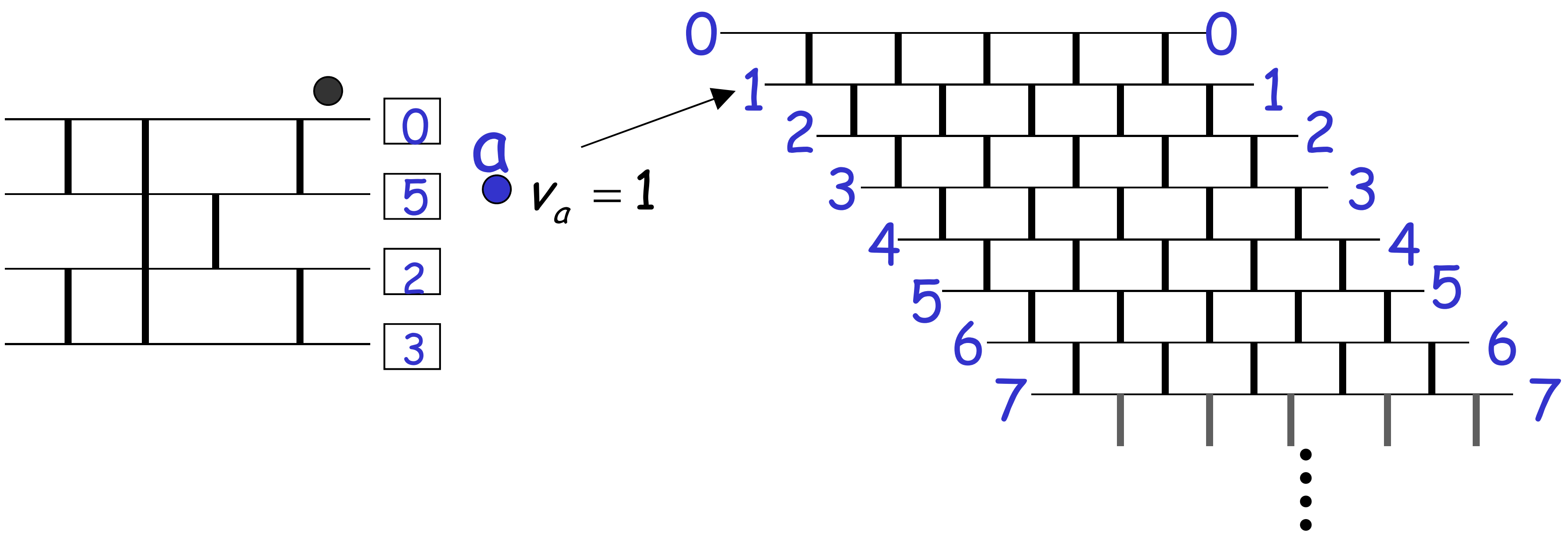


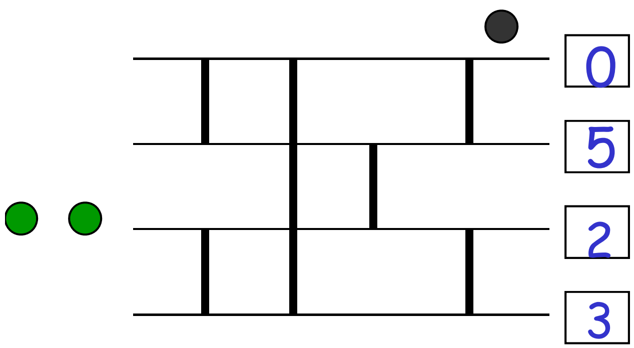


Token
ostaje ovde

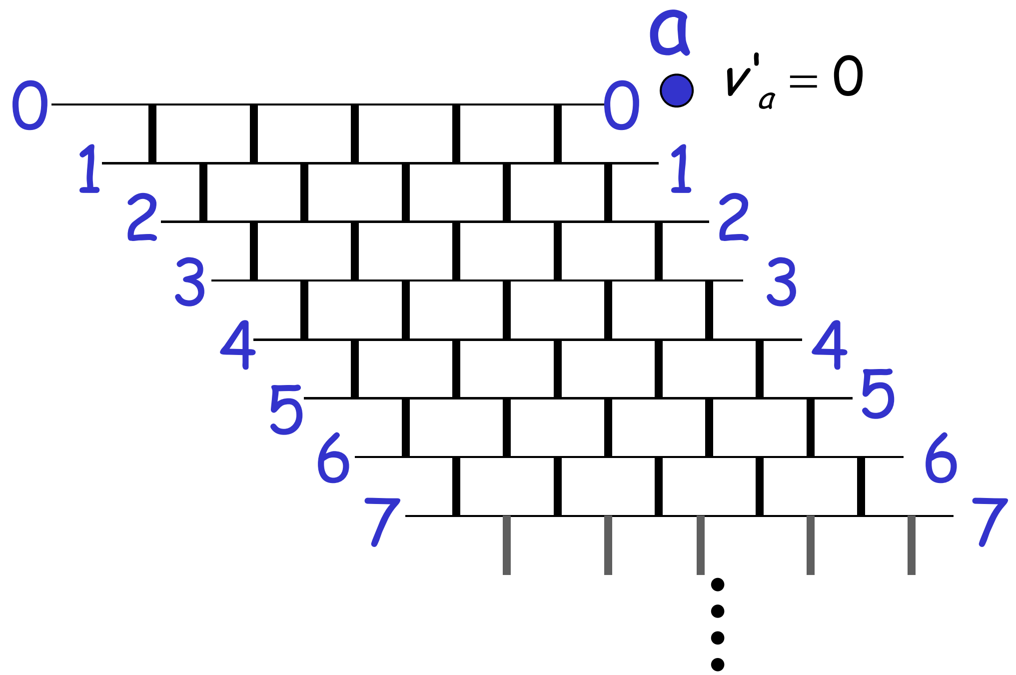




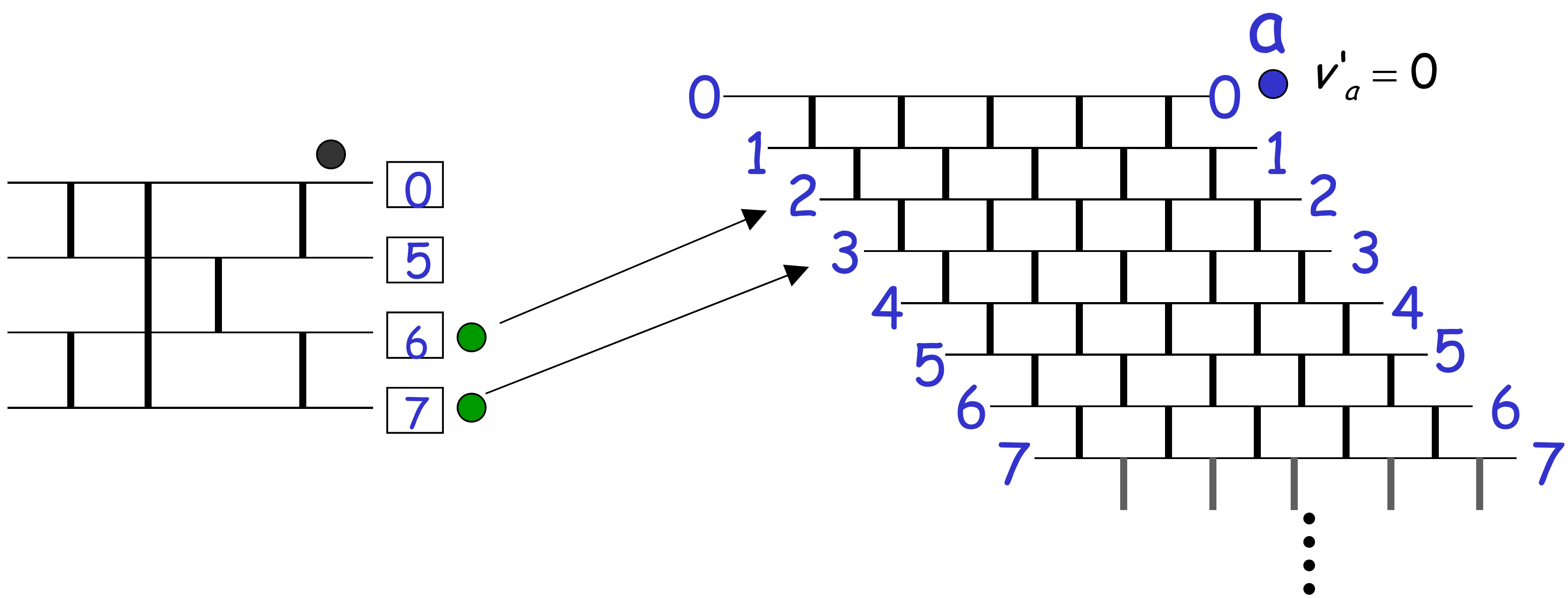

0
5
2
3
a
$v_a = 1$
0 0
1 1
2 2
3 3
4 4
5 5
6 6
7 7

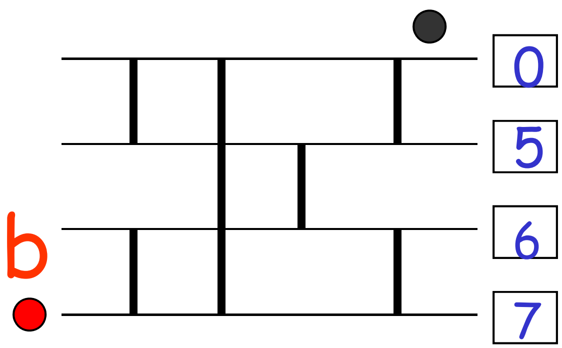


0
5
2
3

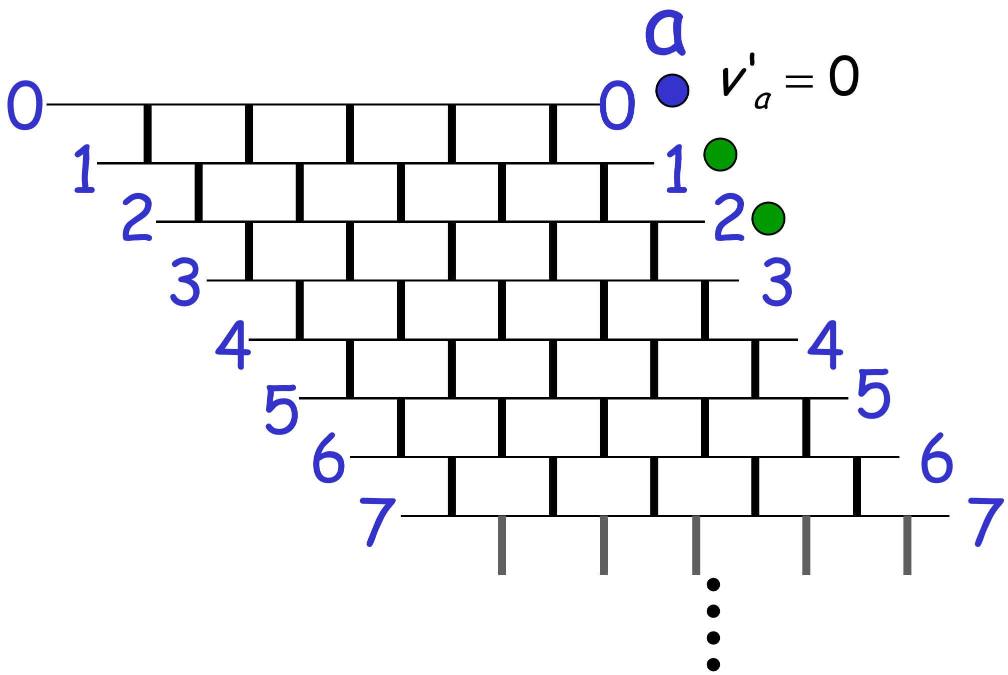


a
$v'_a = 0$
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

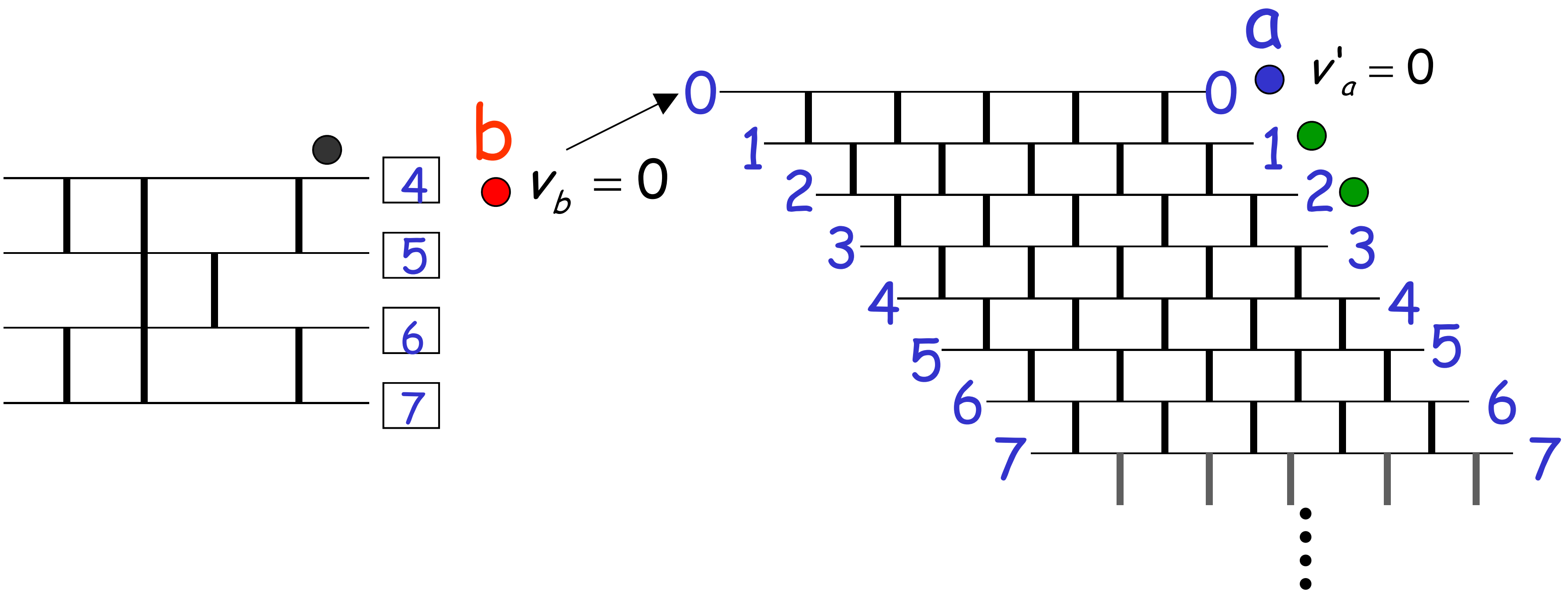

0
5
6
7
a
$v'_a = 0$
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7

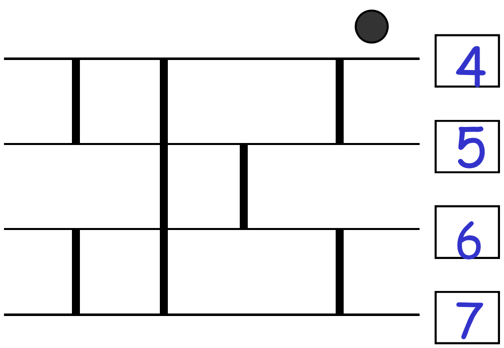


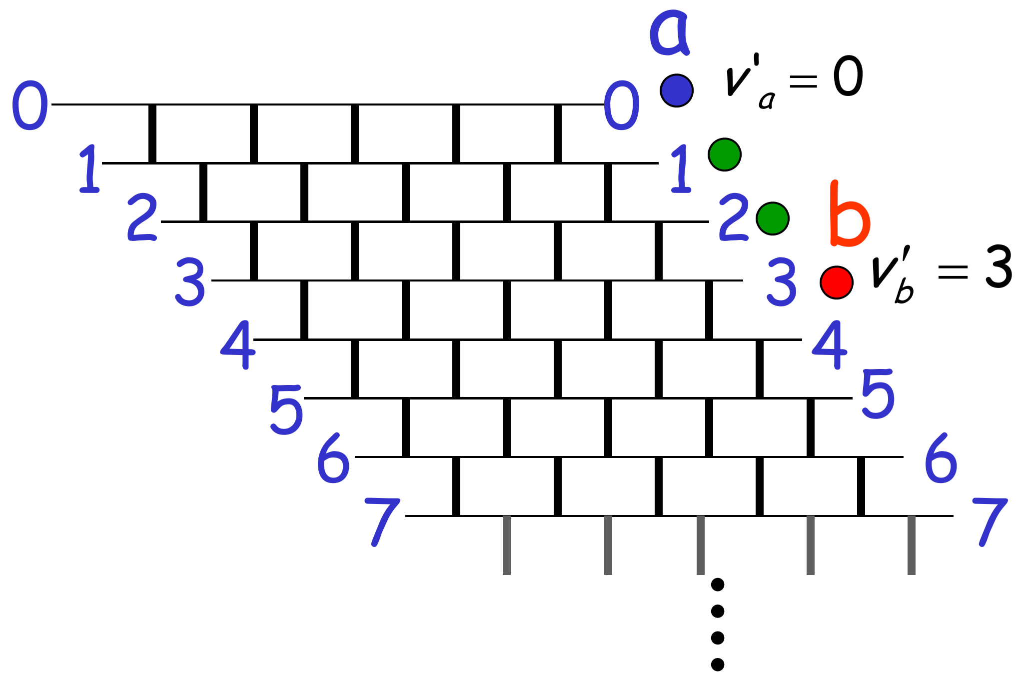
b
0
5
6
7

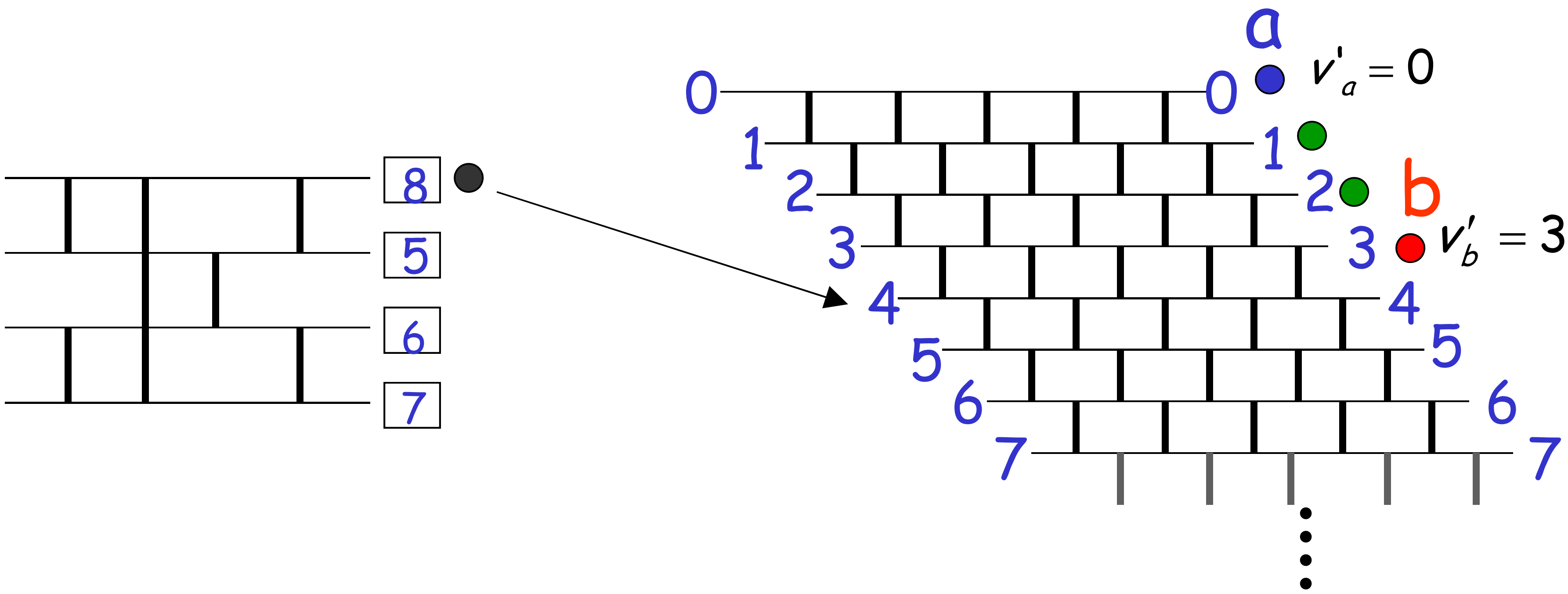


a
$v'_a = 0$
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

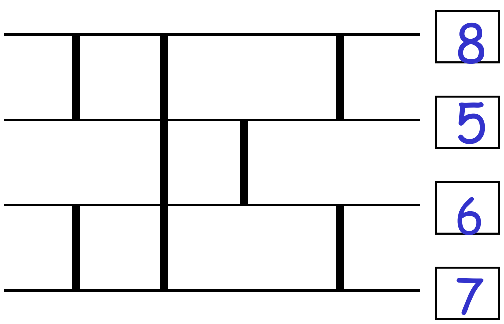

4
5
6
7
b
$v_b = 0$
a
$v'_a = 0$
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

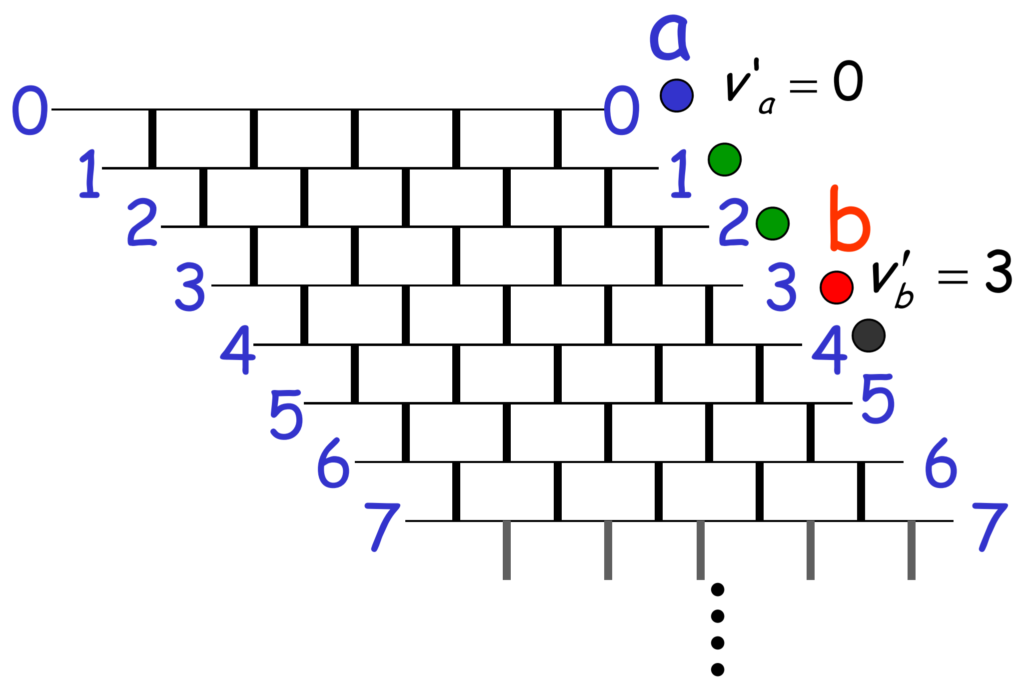

4
5
6
7


a
$v'_a = 0$
b
$v'_b = 3$
0
1
2
3
4
5
6
7


8
5
6
7
a
$v'_a = 0$
b
$v'_b = 3$
0
1
2
3
4
5
6
7

Važi da je: $t^{a}_{exit} < t^{b}_{entry}$ i $v'_a = 0 < 3 = v'_b$



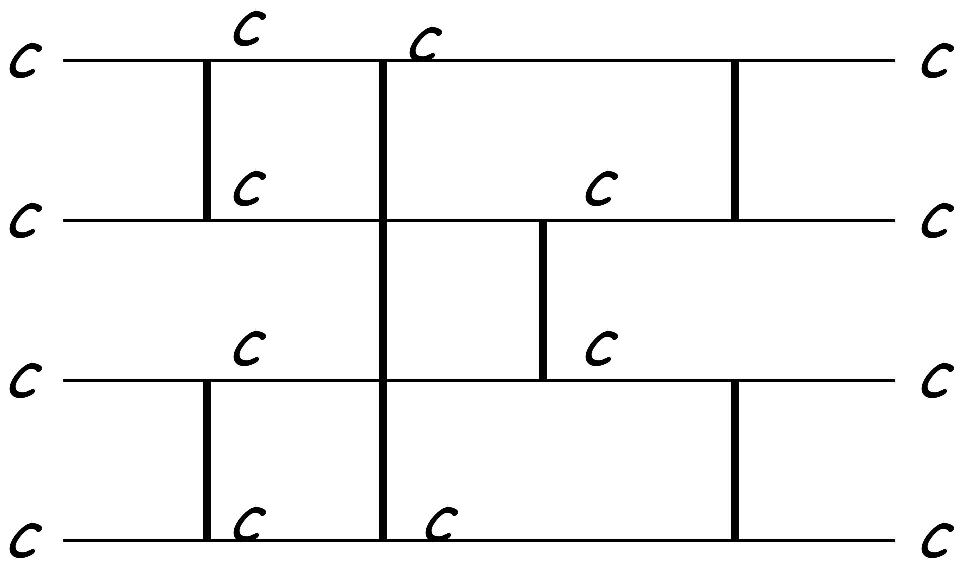


Mogućnost linearizacije je očuvana

Formalno ćemo pokazati
da kosa mreža
zadovoljava mogućnost lienarizacije

Prvo se dokazuje potreban uslov

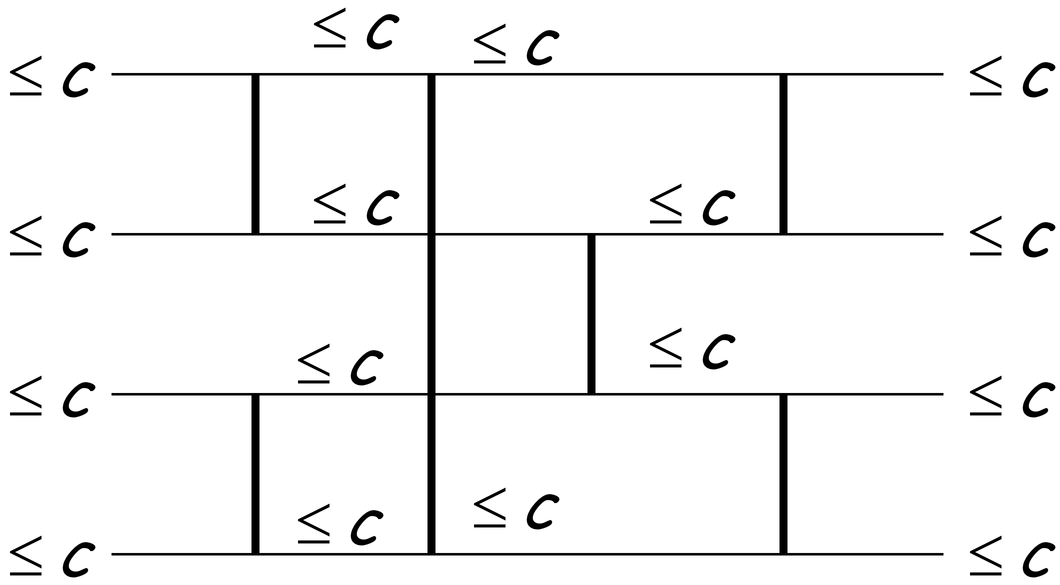
Broj tokena



Za bilo koju mrežu za brojanje:

ako je broj tokena na svakoj ulaznoj žici isti (na primer c), onda će bilo koja žica imati taj isti broj tokena

Broj tokena



Za bilo koju mrežu za brojanje :

ako je broj tokena na svakoj ulaznoj žici ograničen sa c onda će bilo koja žica imati najviše c tokena

Teorema:

Kosa mreža očuvava mogućnost linearizacije

Dokaz:

Dokaz je indukciom na
tokene koji uđu na k žica

Indukciona osnova

Kosa mreža očuvava mogućnost linearizacije za token koji ulazi na žici 0

Trivijalno tačno.

Indukciona hipoteza:

Predpost. da kosa mreža očuvava mogućnost linearizacije za sve tokene koji ulaze do žice $k-1$

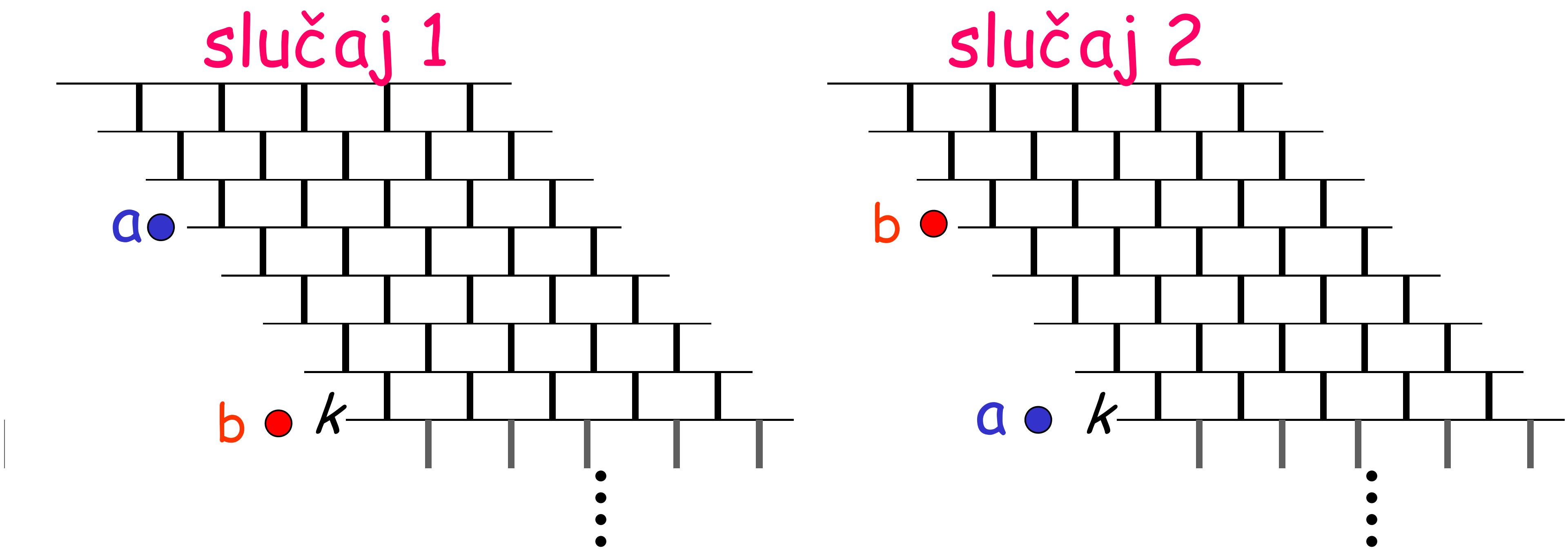
Indukcioni korak:

Pokazaćemo da kosa mreža očuvava mogućnost linearizacije za sve tokene koji ulaze na prvih k žica

Predpost. da postoje dva tokena a i b koji narušavaju mogućnost linearizacije:

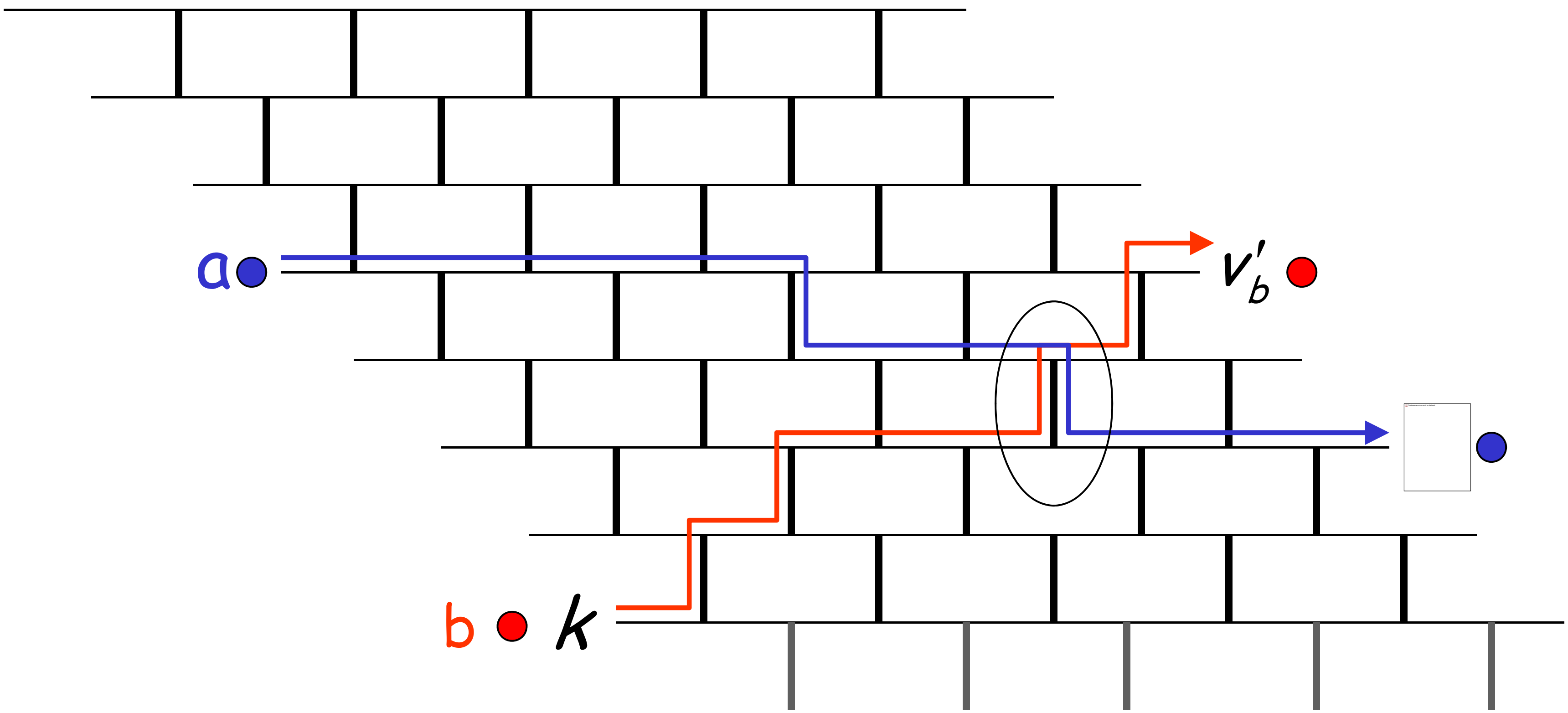
$$t_{exit}^{a} < t_{entry}^{b} \qquad v_a' > v_b'$$

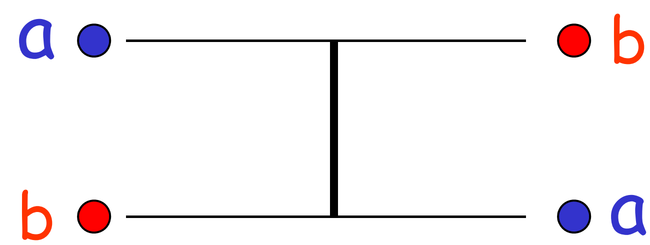
Postoje dva moguća slučaja:



$$t_{exit}^{a} < t_{entry}^{b} \qquad v_a' > v_b'$$

SLUČAJ 1





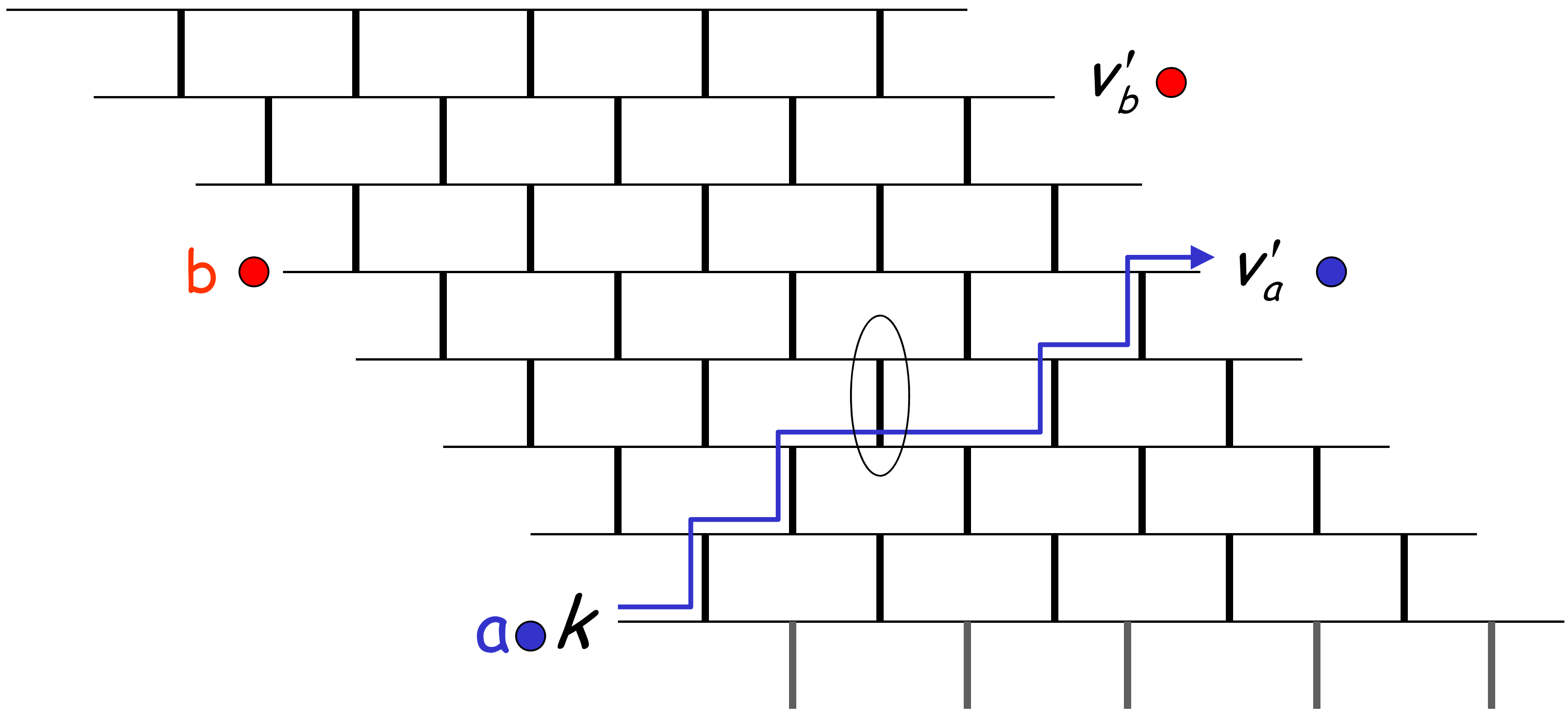
Token a ulazi prvi i zato mora biti

Nemoguće!!!

(pošto ni jedan drugi token ne ulazi u balanser)

SLUČAJ 2

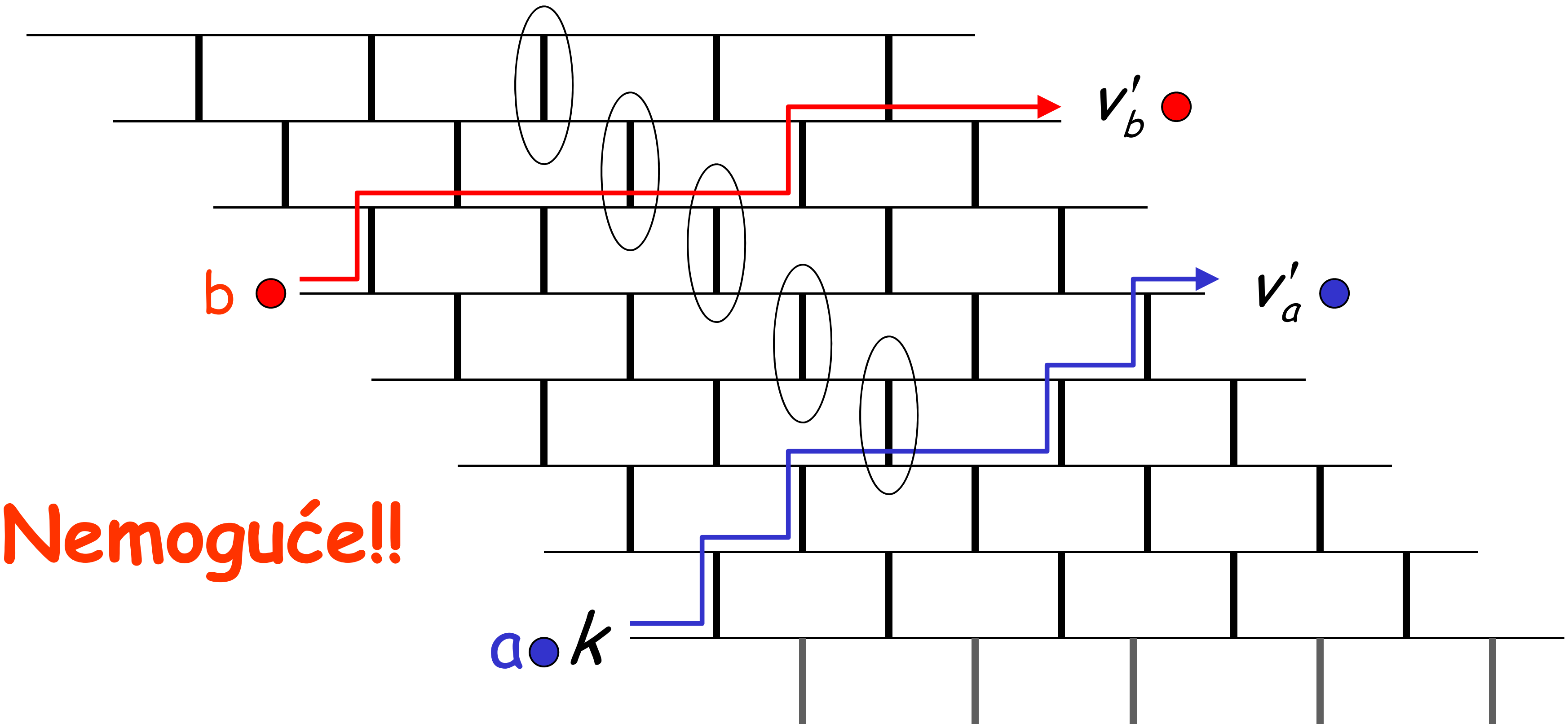
$t^{a}_{exit} < t^{b}_{entry}$ $\quad v'_a > v'_b$



pod-slučaj:

token **a** izlazi na južnoj žici nekog balansera

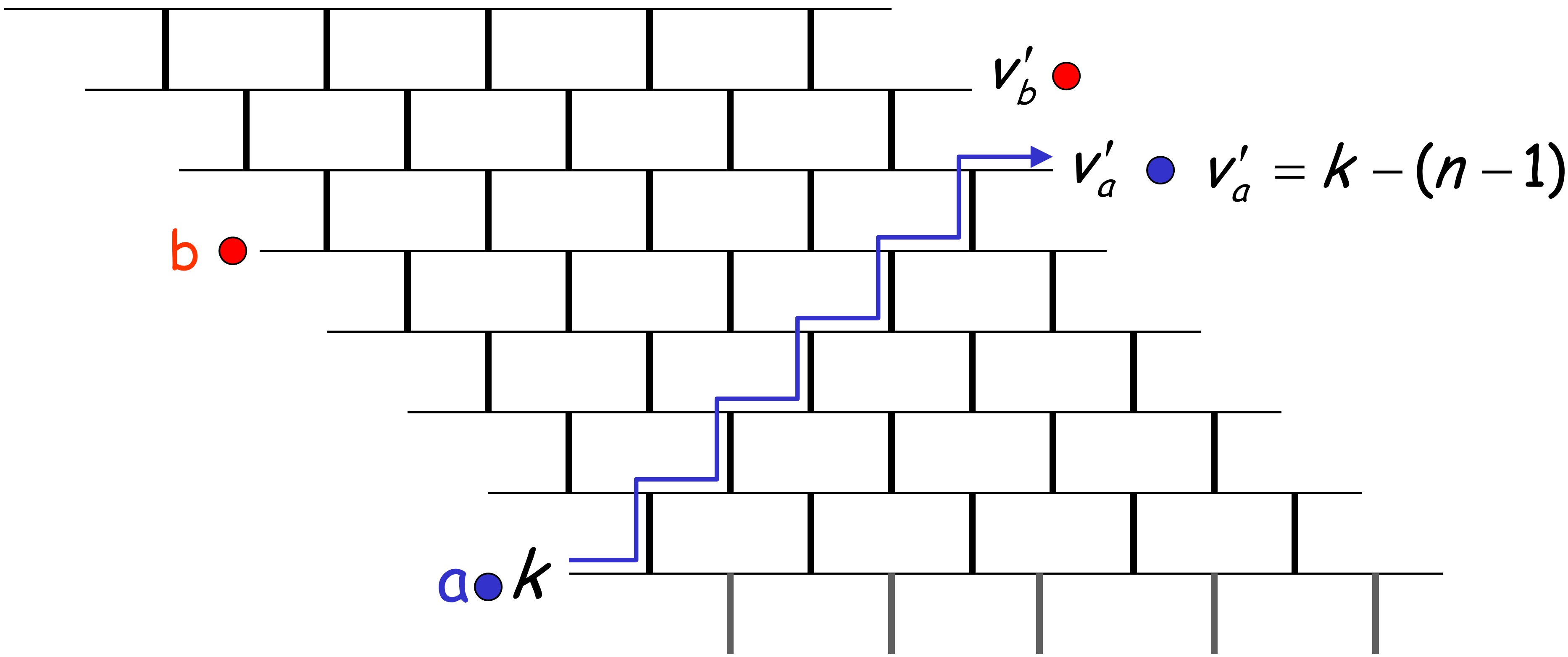
$$t_{exit}^{a} < t_{entry}^{b} \qquad v_a' > v_b'$$



Nemoguće!!

Pošto bi svi ovi balanseri morali biti iskorišćeni od tokena koji su ušli pre nego je token a izašao

$$t_{exit}^{a} < t_{entry}^{b} \qquad v'_a > v'_b$$



Token a mora izaći sa severne žice svakog balansera

Tvrdnja:

Bar $v'_a = k - (n-1)$ tokena mora
preći kombinovanu mrežu
pre nego je token a ušao u kosu mrežu

(različiti od a)

Ukupno tokena kad token a dobije svoju vrednost v_a od prve mreže za brojanje

$$x = x_1 + x_2$$

Tokeni u kombinovanoj mreži

Tokeni već izašli iz kombinovane mreže

$$x \geq k$$

važi jer inače mreža za brojanje ne bi brojala

$$x_1 \leq n - 1$$

važi jer postoji najviše n-1 procesa (isključujući a)

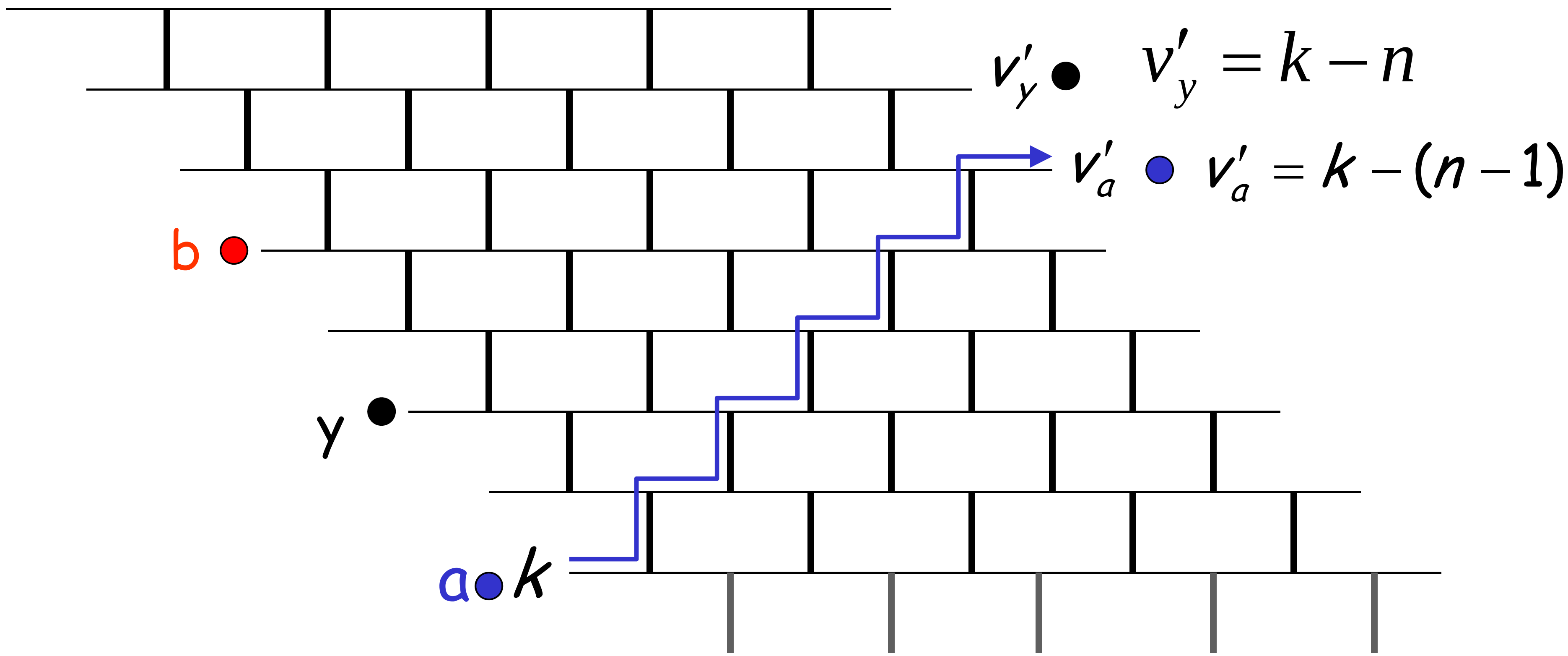
$$x = x_1 + x_2 \Rightarrow$$

$$x_2 = x - x_1 \geq k - (n - 1)$$

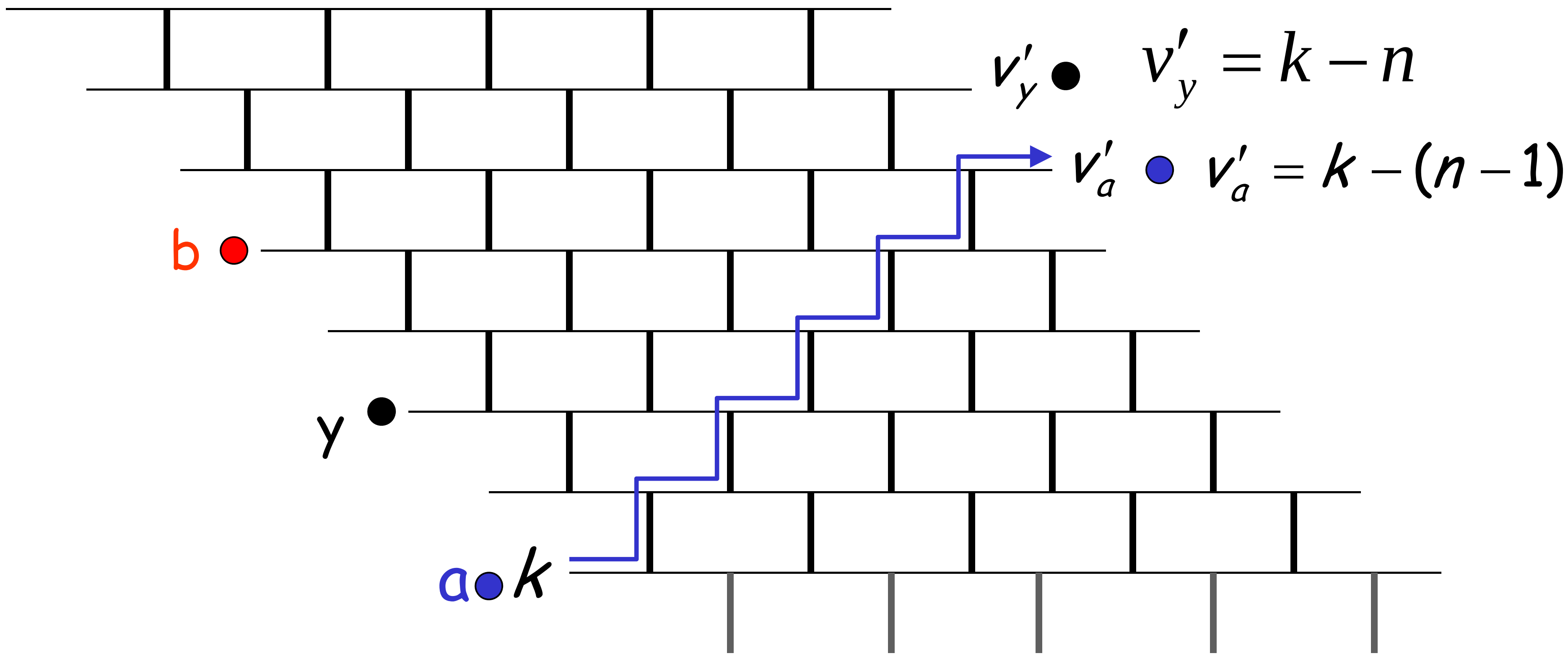
Mora biti da je bar $k-(n-1)$ tokena (iz skupa x_2) primilo vrednosti, od prve mreže za brojanje, koje su manje od k

Ovo važi, jer inače kada bi zaustavili izvršenje sa tokenom a prva mreža za brojanje ne bi dala sve vrednosti između 1 i k

Zato, postoji bar
jedan takav token (na primer token y)
za koji je: $v'_y = k - n$



Pošto se, na osnovu indukcione hipoteze, mreža može linearizovati u prvih k-1 žica: $v'_y < v'_b$ (b ulazi posle y)



Sledi da je: $v'_y < v'_b$ i $v'_a > v'_b$

Nemoguće!!!