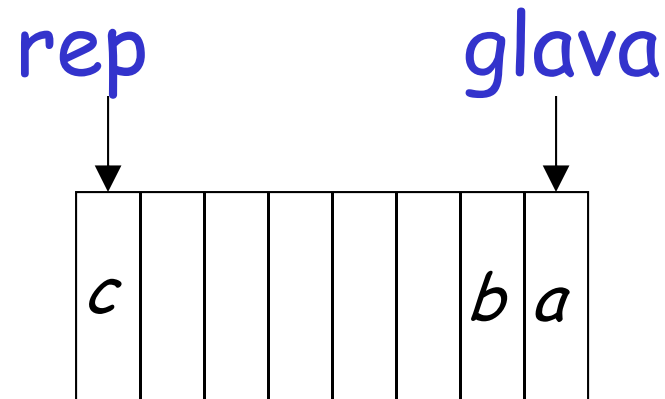


Hijerahija konsenzusa

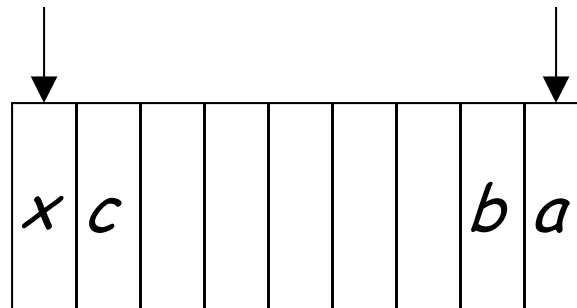
Deo 2

FIFO (Red)

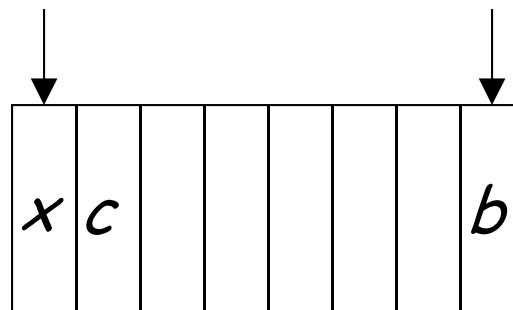
FIFO Objekt Q



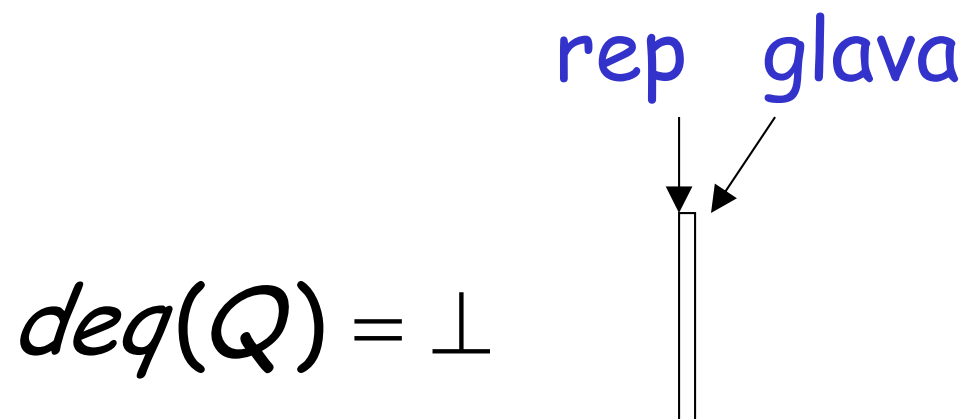
$enq(Q, x)$



$deq(Q) = a$



Poseban slučaj: prazan red



Teorema: FIFO objekt ima broj konsenzusa 2

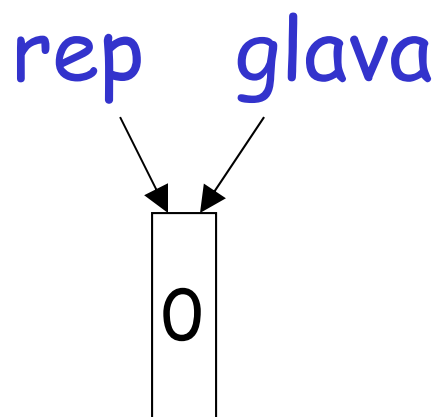
Dokaz:

1. Možemo rešiti WF (wait-free) konsenzus korišćenjem FIFO (i read/write) objekata za 2 procesa
2. **Ne možemo** rešiti WF konsenzus korišćenjem FIFO (i read/write) objekata za 3 ili više procesa

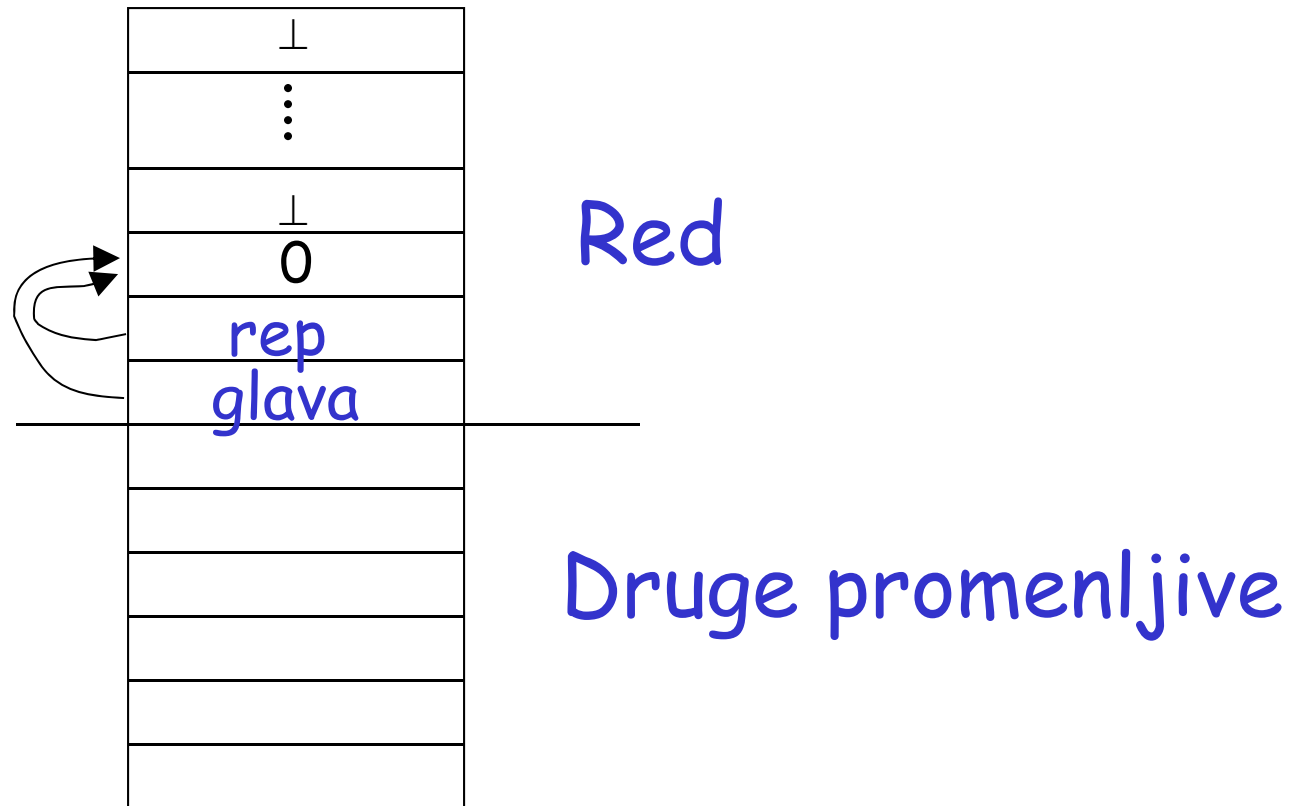
Dokaz - Deo 1

Jedan algoritam WF konsenzusa
za 2 proc korišćenjem FIFO objekta
(i read/write objekata)

Na početku:



Deljena memorija



Lokalne promenljive

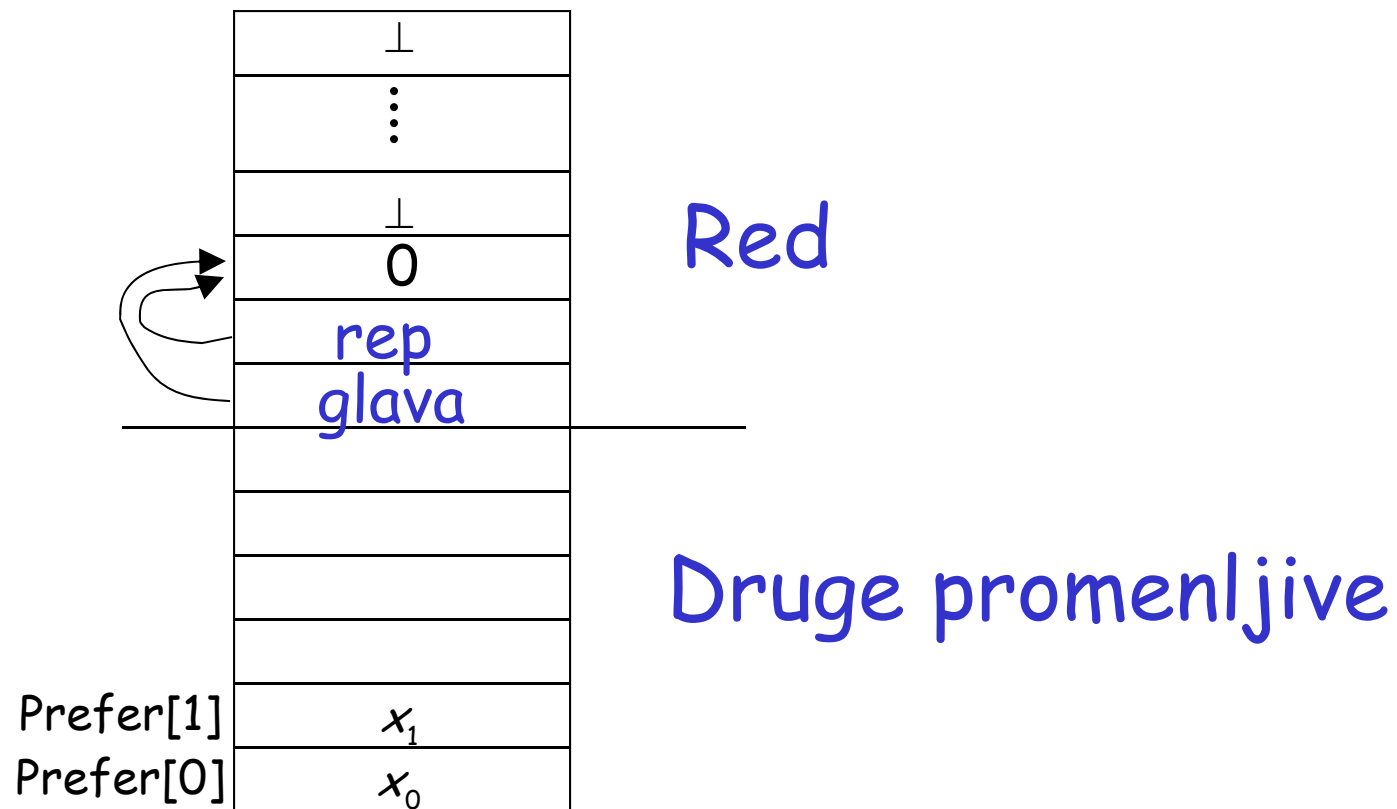
$$p_0 \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \qquad \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} p_1$$

$$x_i, y_i \in \{0,1\}$$

x_i : početne vred za problem konsenzusa

y_i : vrednosti rezultata za problem konsenzusa

Deljena memorija



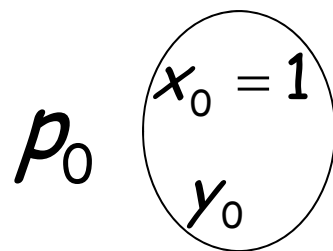
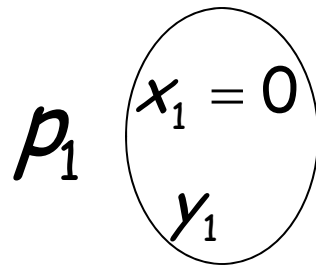
Početne vrednosti

Kod za procesor p_i $i \in \{0,1\}$

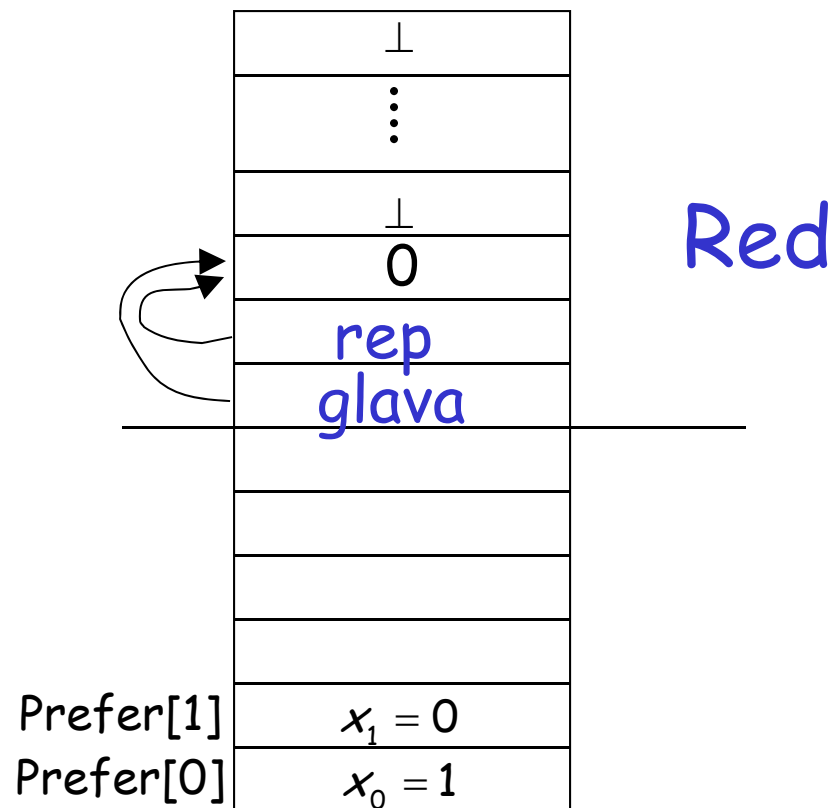
```
prefer[i]  $\leftarrow x_i$   
If  $deq(Q) == 0$  //da li sam ja prvi?  
    then  $y_i \leftarrow x_i$  //da, izaberi moju vred  
    else  $y_i \leftarrow prefer[1-i]$  //ne, izaberi vred  
                                             drugog procesora
```

Napomena: ovaj algoritam koristi FIFO objekt
i read/write objekte

Primer izvršenja:

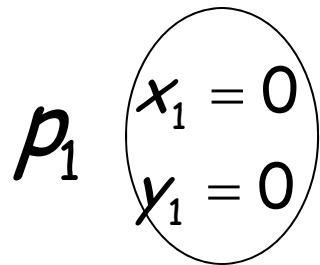


Deljena memorija



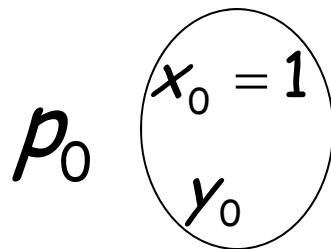
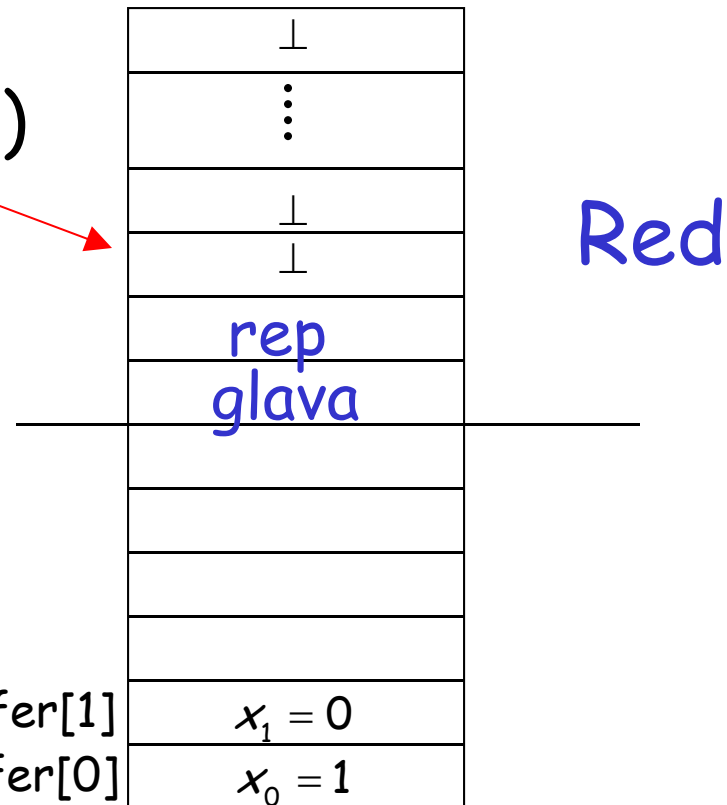
Predpost. da p_1 prvi pristupa redu

Deljena memorija



$0 = \text{deg}(Q)$

odluči se za
svoju vrednost



Predpost. da p_2 drugi pristupa redu

Deljena memorija

$$p_1 \begin{pmatrix} x_1 = 0 \\ y_1 = 0 \end{pmatrix}$$

Konsenzus postignut

$$\perp = \text{deq}(Q)$$

$$p_0 \begin{pmatrix} x_0 = 1 \\ y_0 = 0 \end{pmatrix}$$

Prefer[1]
Prefer[0]

\perp
\vdots
\perp
\perp
rep
glava
$x_1 = 0$
$x_0 = 1$

Red

odluči se za vred. drugog procesora

Dokaz - Deo 2

Dokazaćemo da:

Ne postoji algoritam WF
konsenzusa koji koristi samo FIFO
i read-write objekte za $n \geq 3$

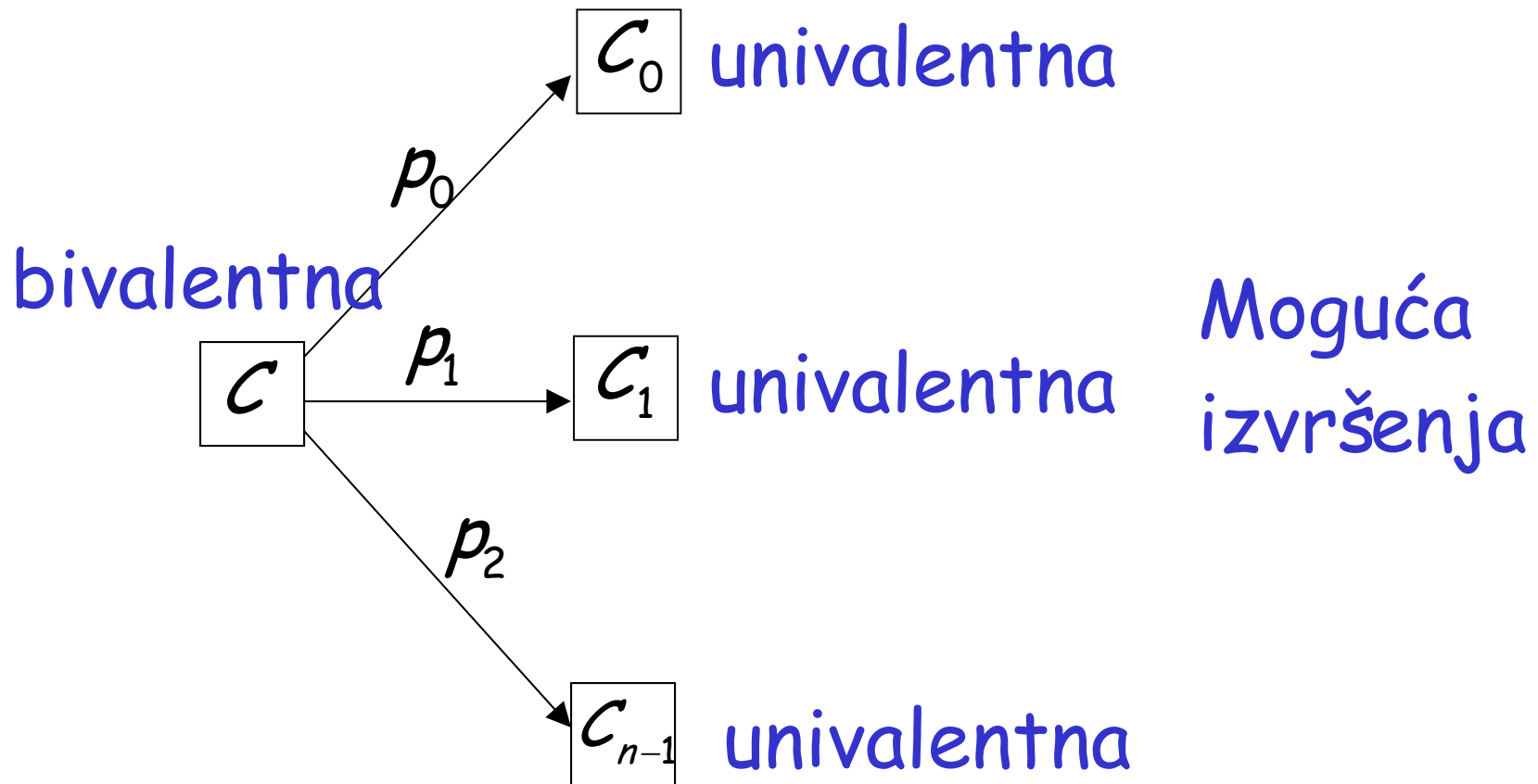
Razmotrimo tri procesora
(isti dokaz se generalizuje za više)

Postoji neka bivalentna početna konfiguracija
(ovo smo dokazali ranije)

Pokazaćemo da svaka bivalentna konfiguracija
ima neki procesor koji nije kritičan

Zbog toga, možemo konstruisati beskonačno
izvršenje sa bivalentnim konfiguracijama gde
se konsenzus nikada ne postiže

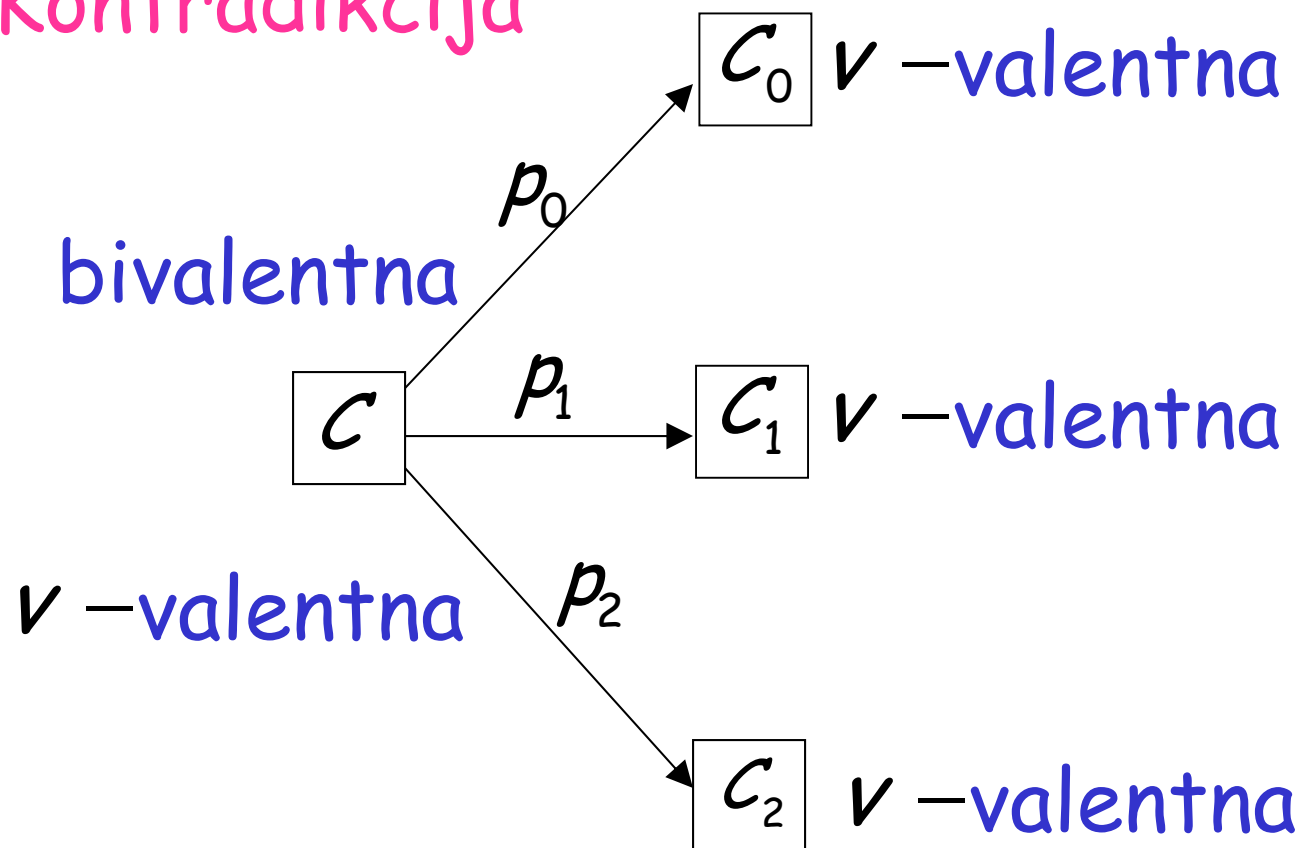
Predpost. radi kontradikcije da su
svi procesori kritični



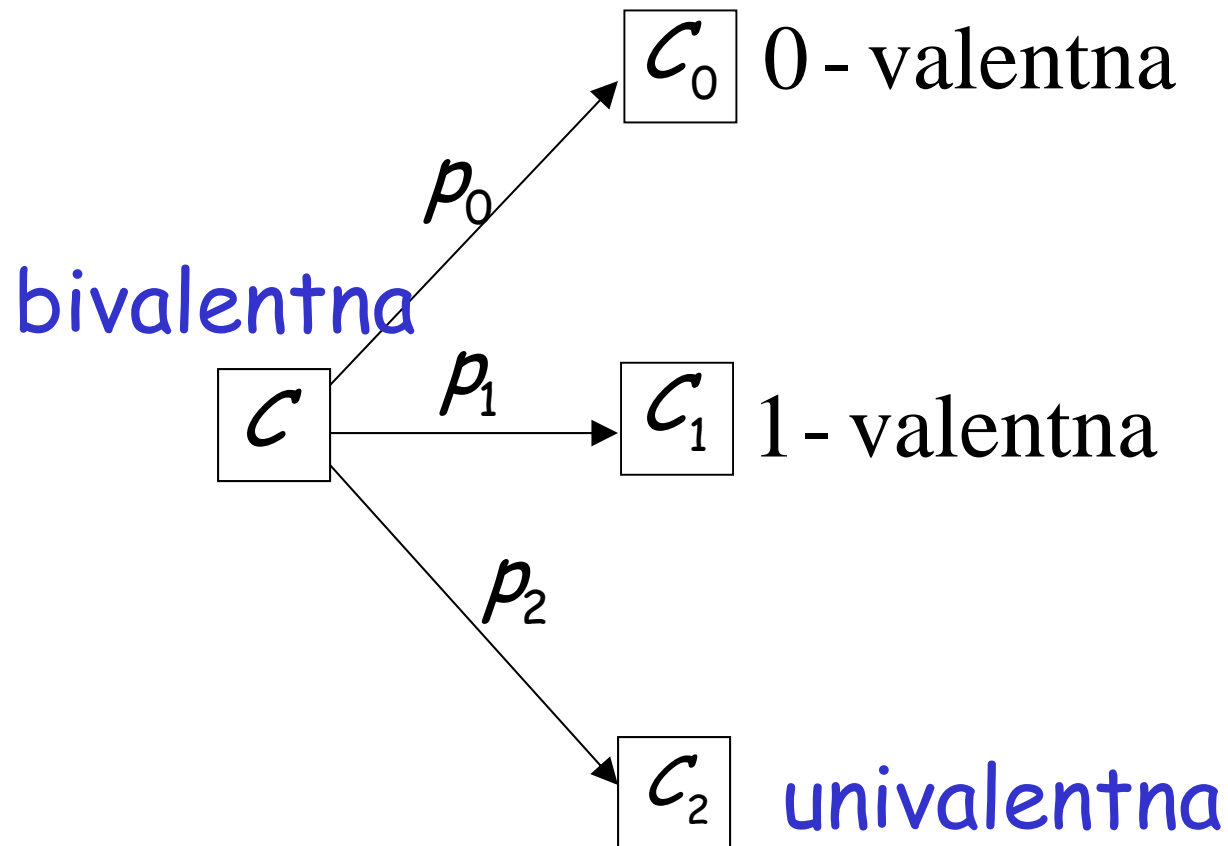
Ne može biti da svi imaju istu valencu

($v = 0$ ili 1)

Kontradikcija



Moraju postojati dva procesora sa različitim valencama

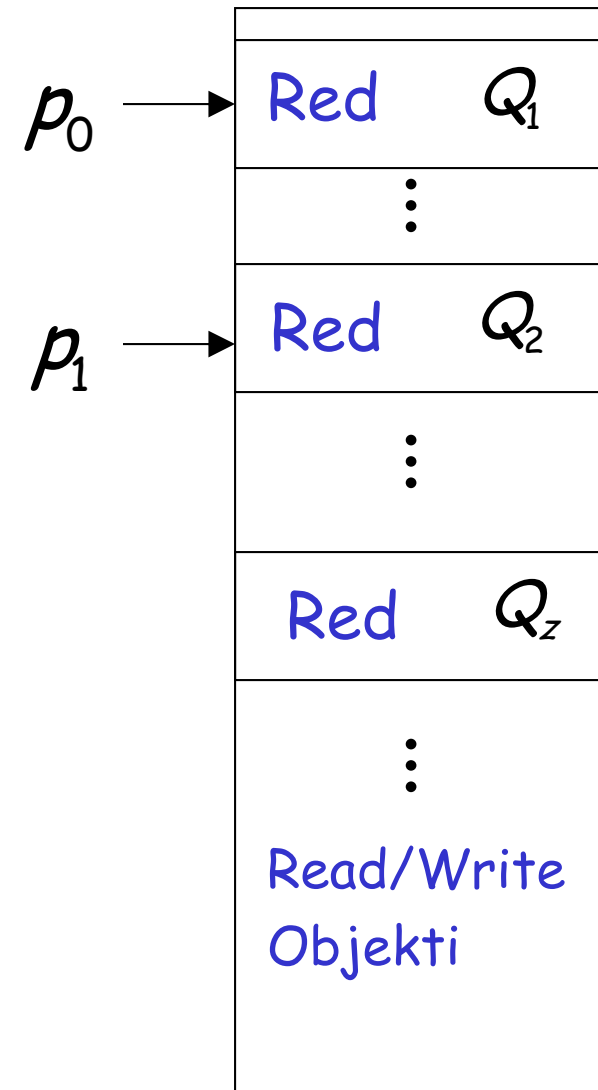
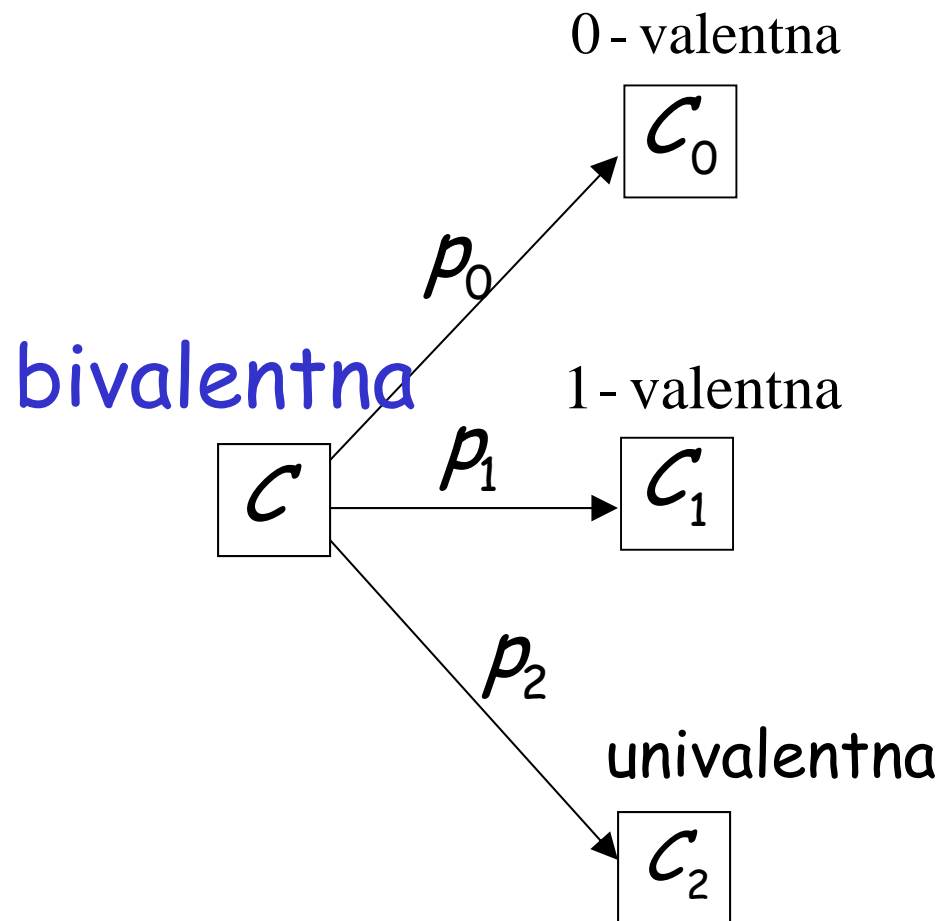


Deljena memorija

Red	Q_1
⋮	
Red	Q_2
⋮	
Red	Q_z
⋮	
Read/Write Objekti	

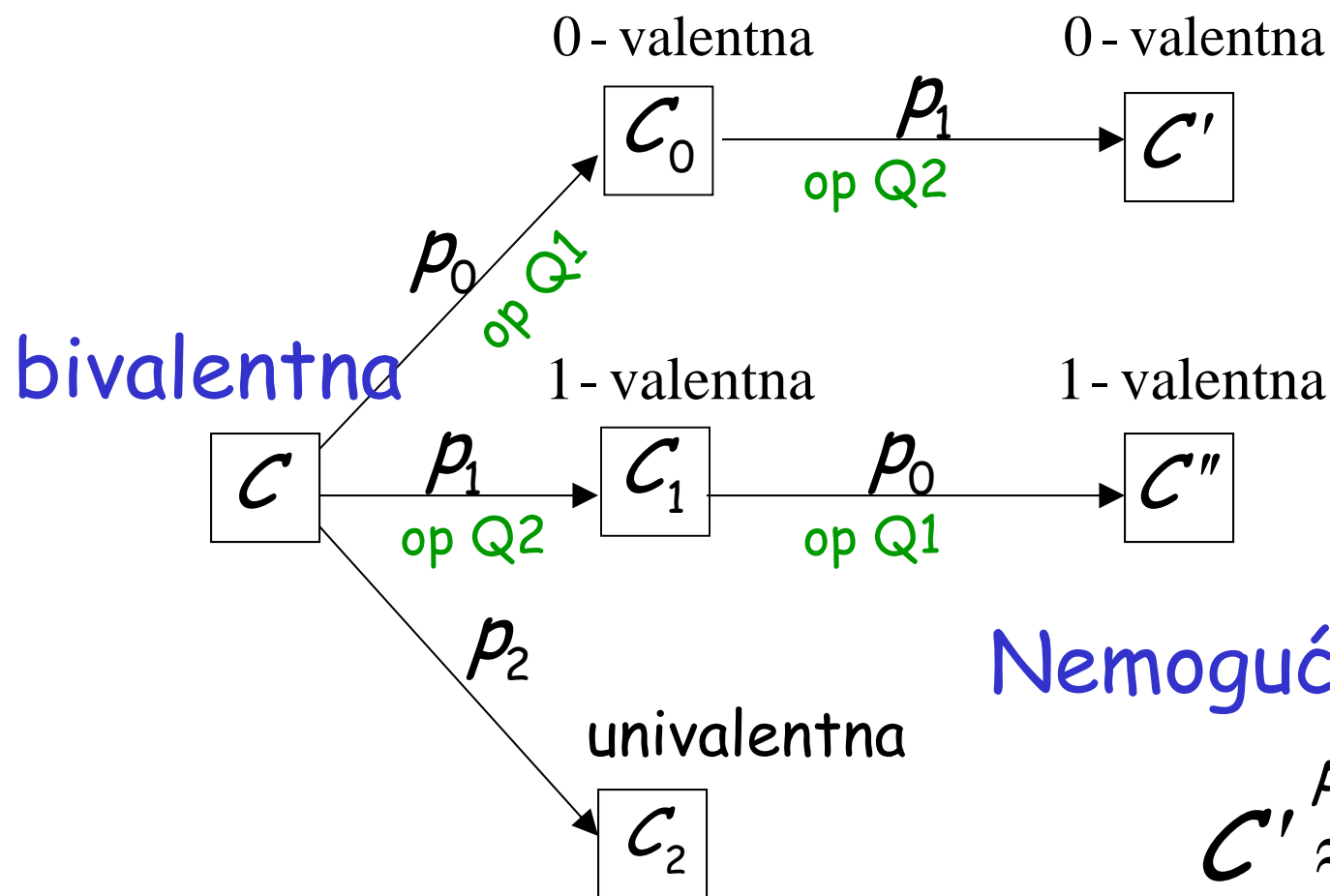
Sluč.: procesori pristupaju različitim objektima

Deljena memorija



Napomena: da je objekt bio čitan/pisan, analiza bi bila slična

Dva moguća izvršenja

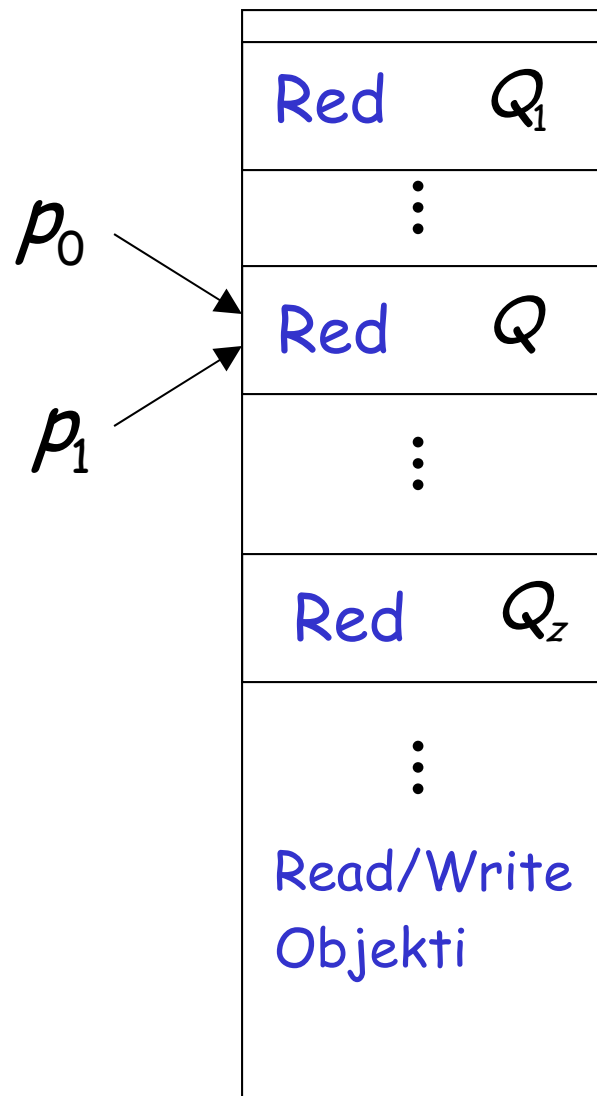
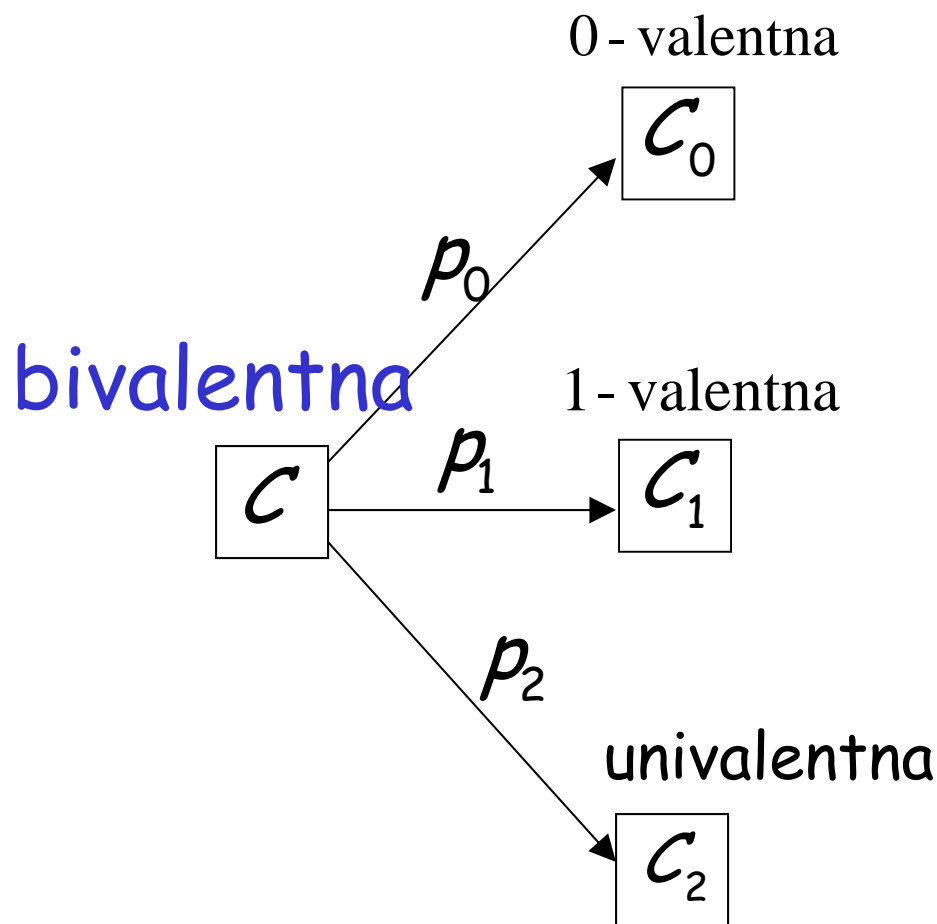


Nemoguće jer je

$$C' \stackrel{p_2}{\approx} C''$$

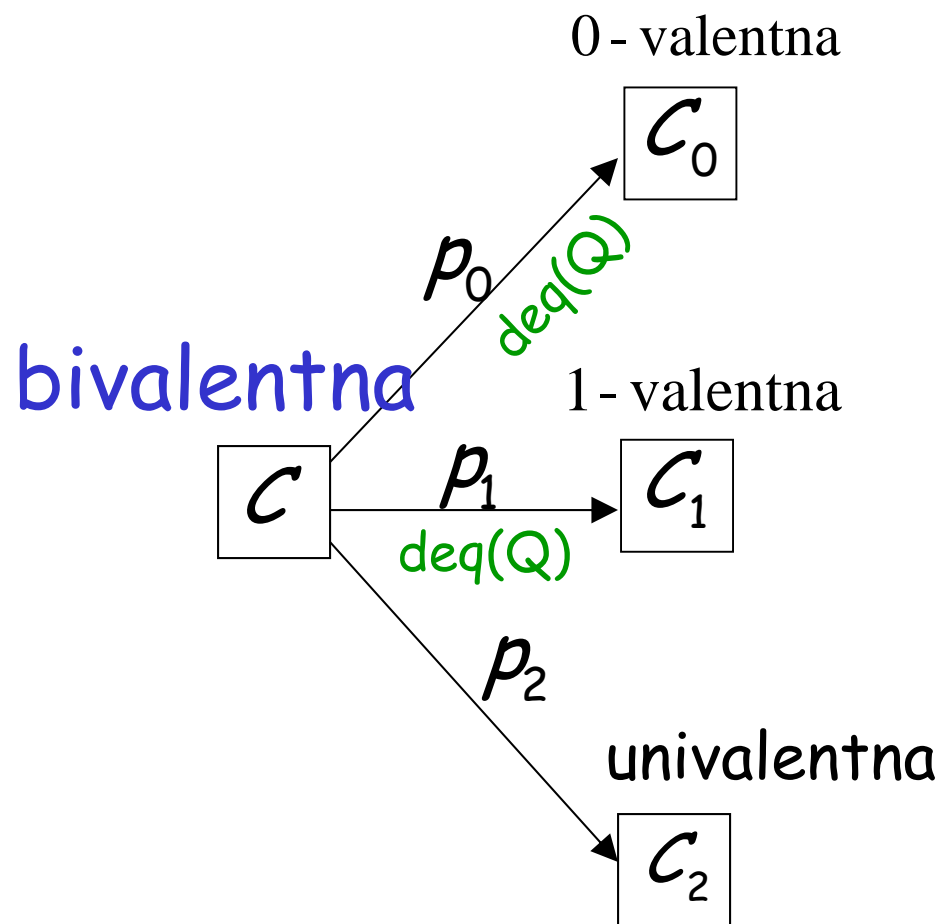
Sluč.: procesori pristupaju
istom objektu

Deljena memorija

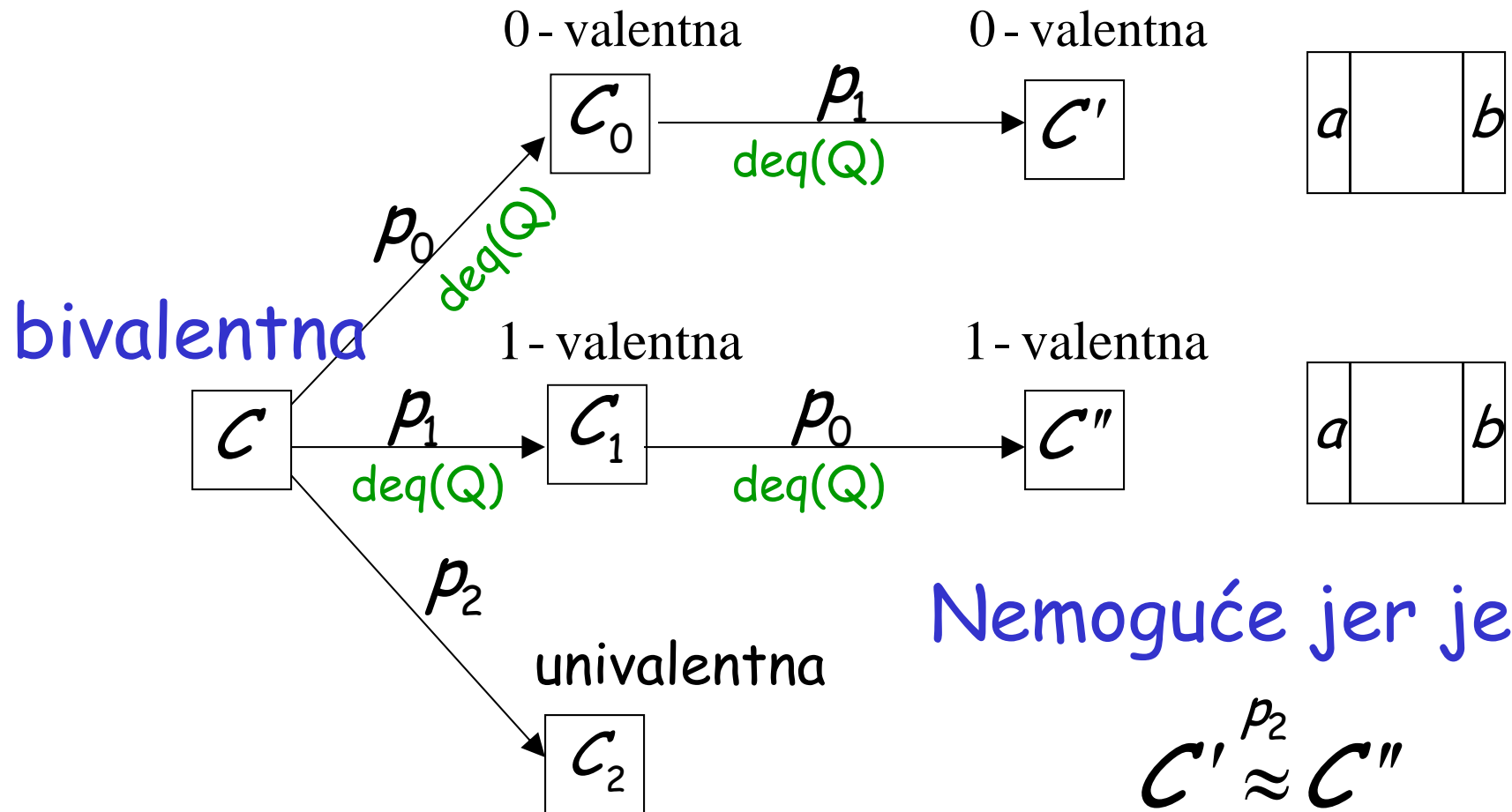
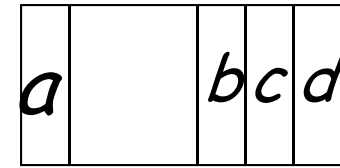


Napomena: da je objekt bio čitan/pisan, analiza bi
bila ista kao u slučaju sa read/write objektima

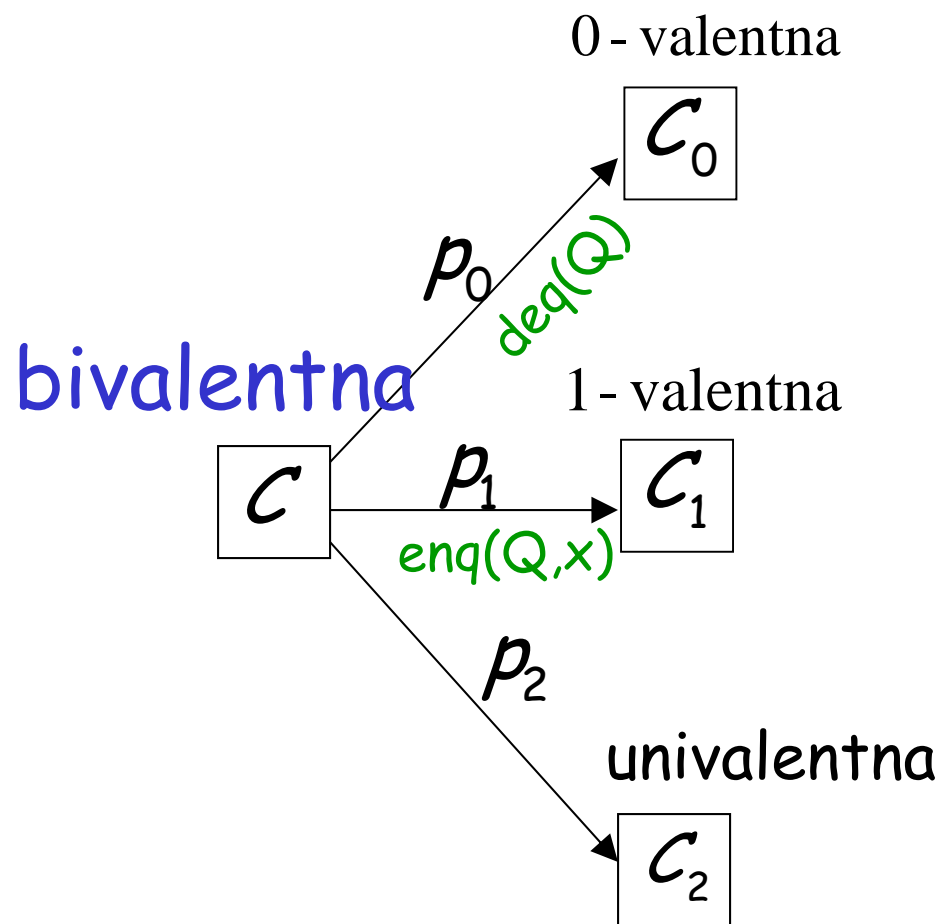
Podslučaj: deq/deq



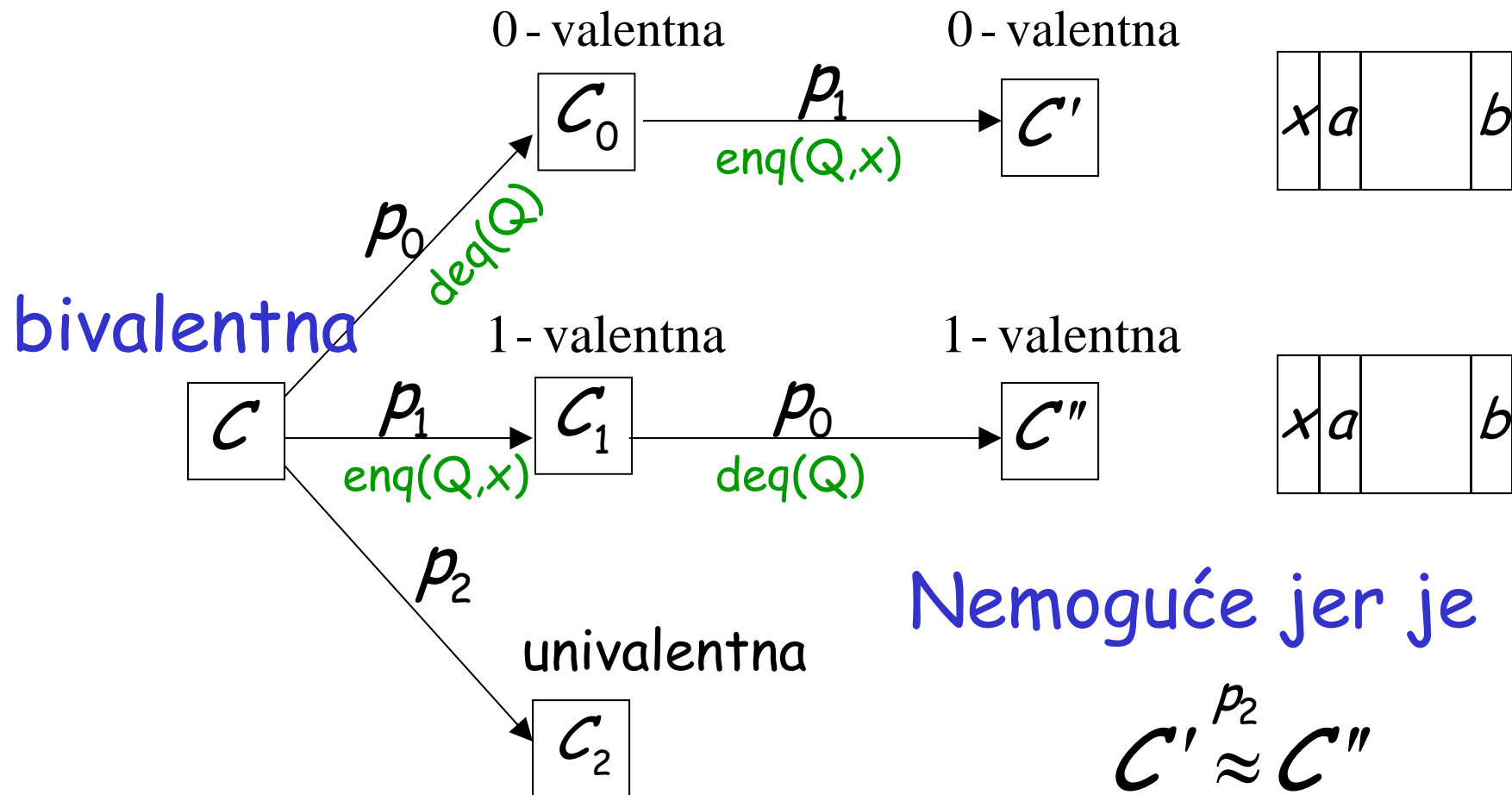
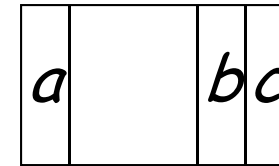
Red Q pre operacija:



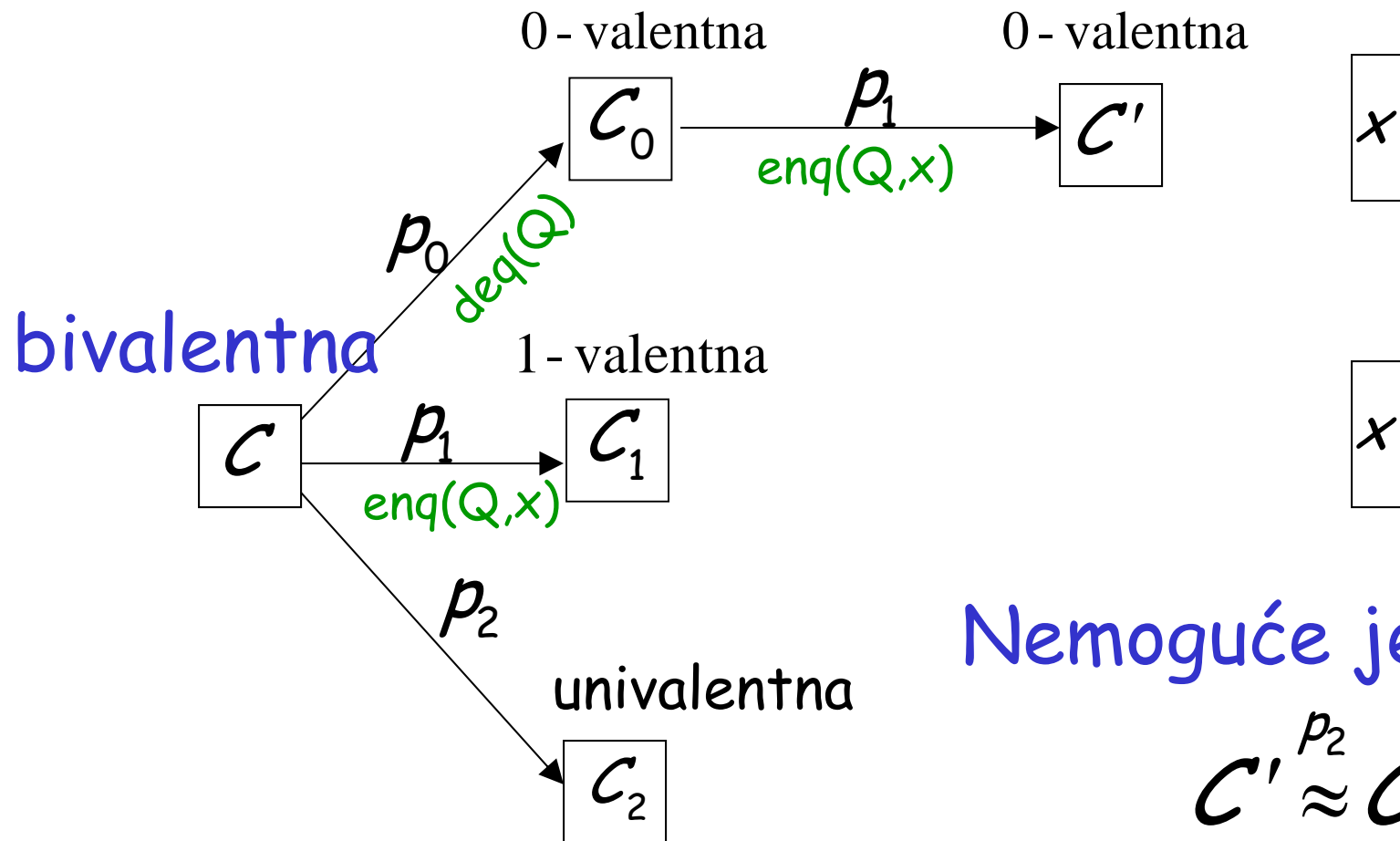
Podslučaj: deq/enq



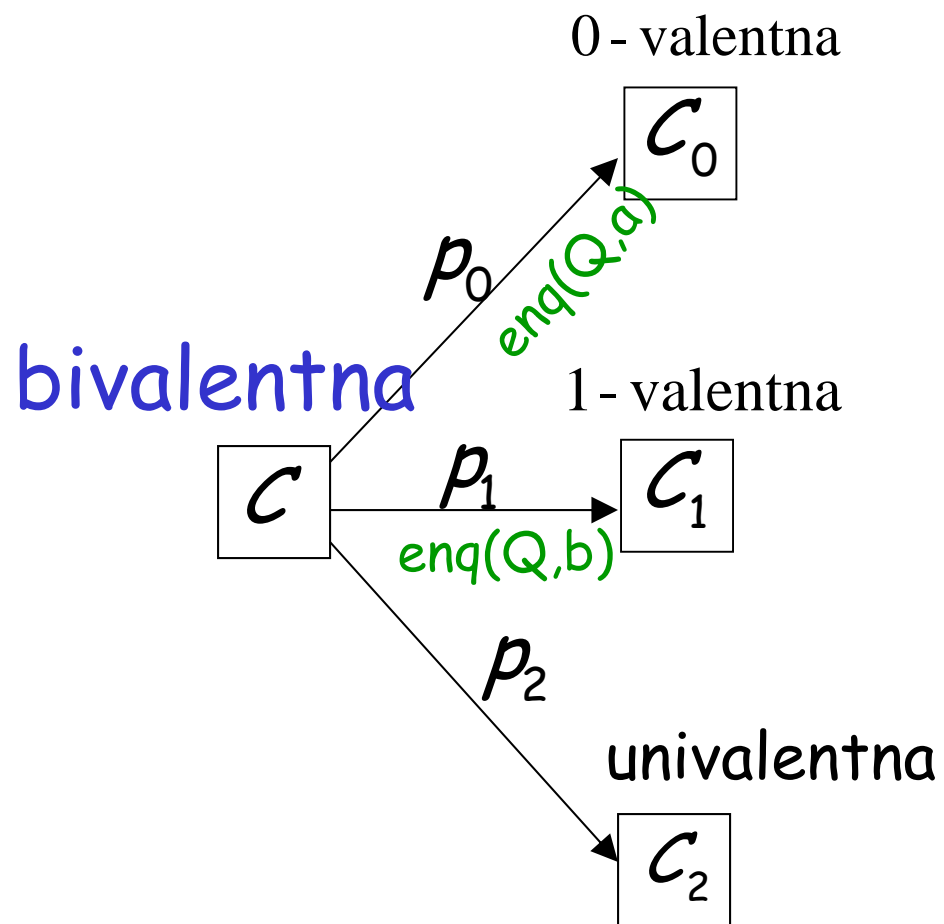
Predpost. Q nije bio prazan



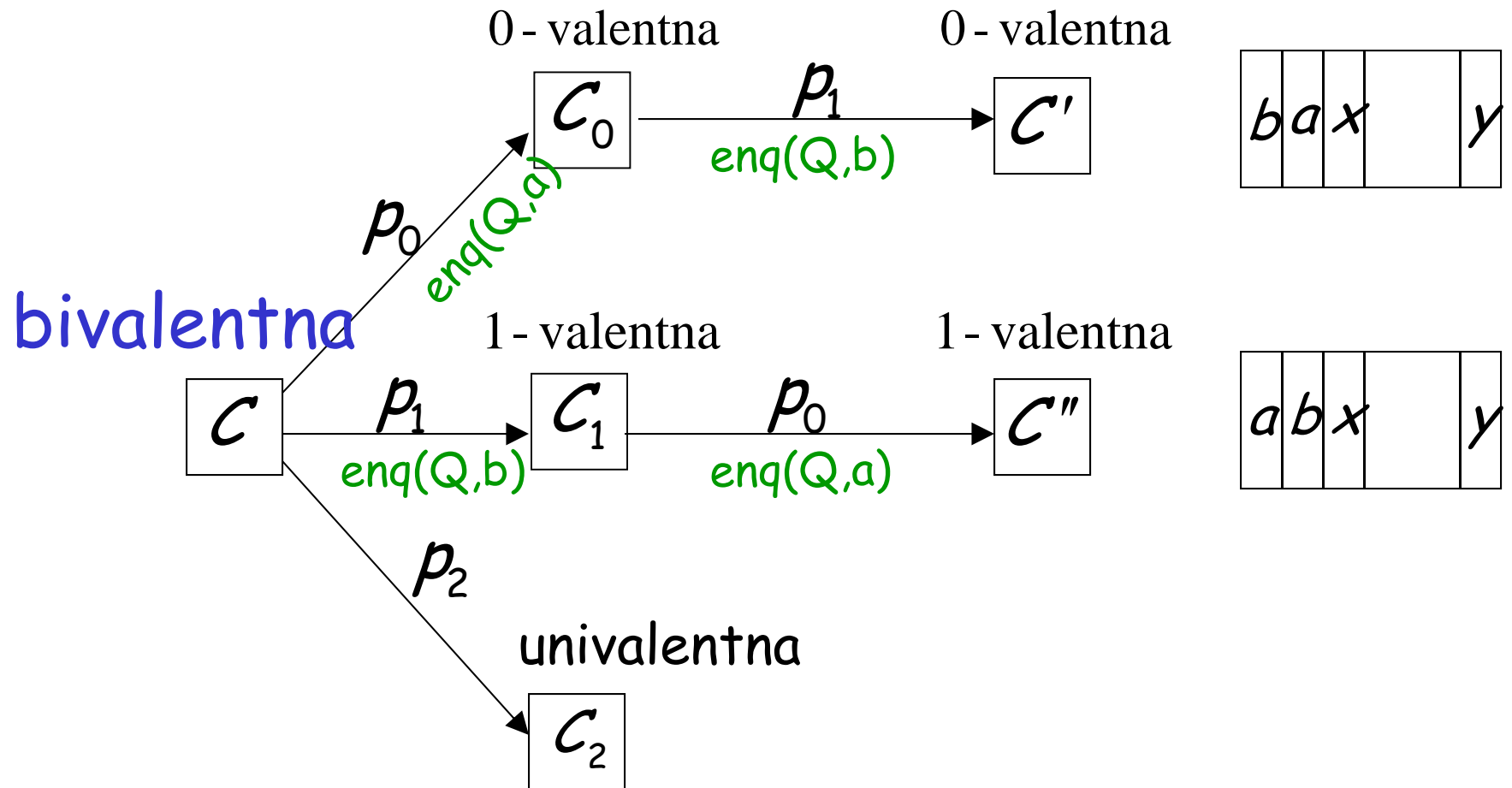
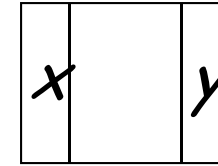
Predpost. da je Q bio prazan

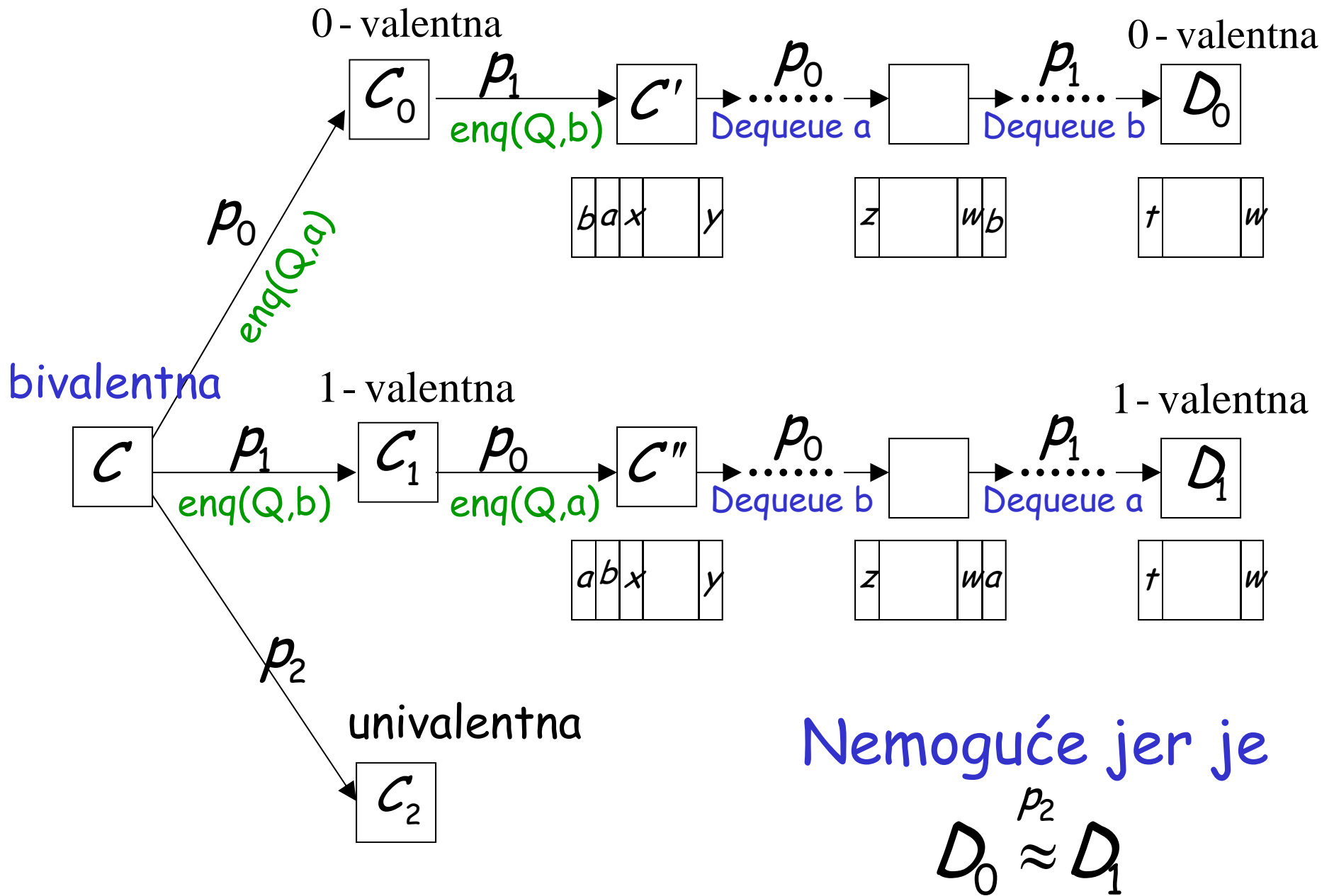


Podslučaj: enq/enq

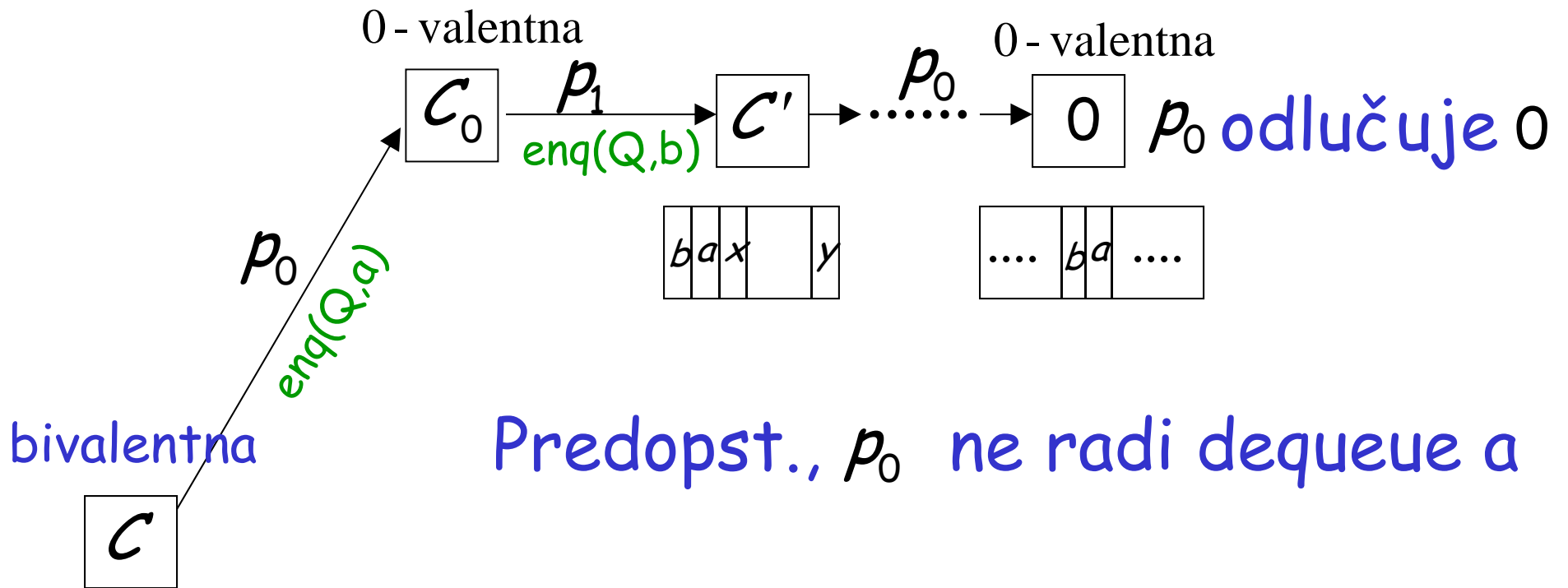


Predpost. Q nije bio prazan



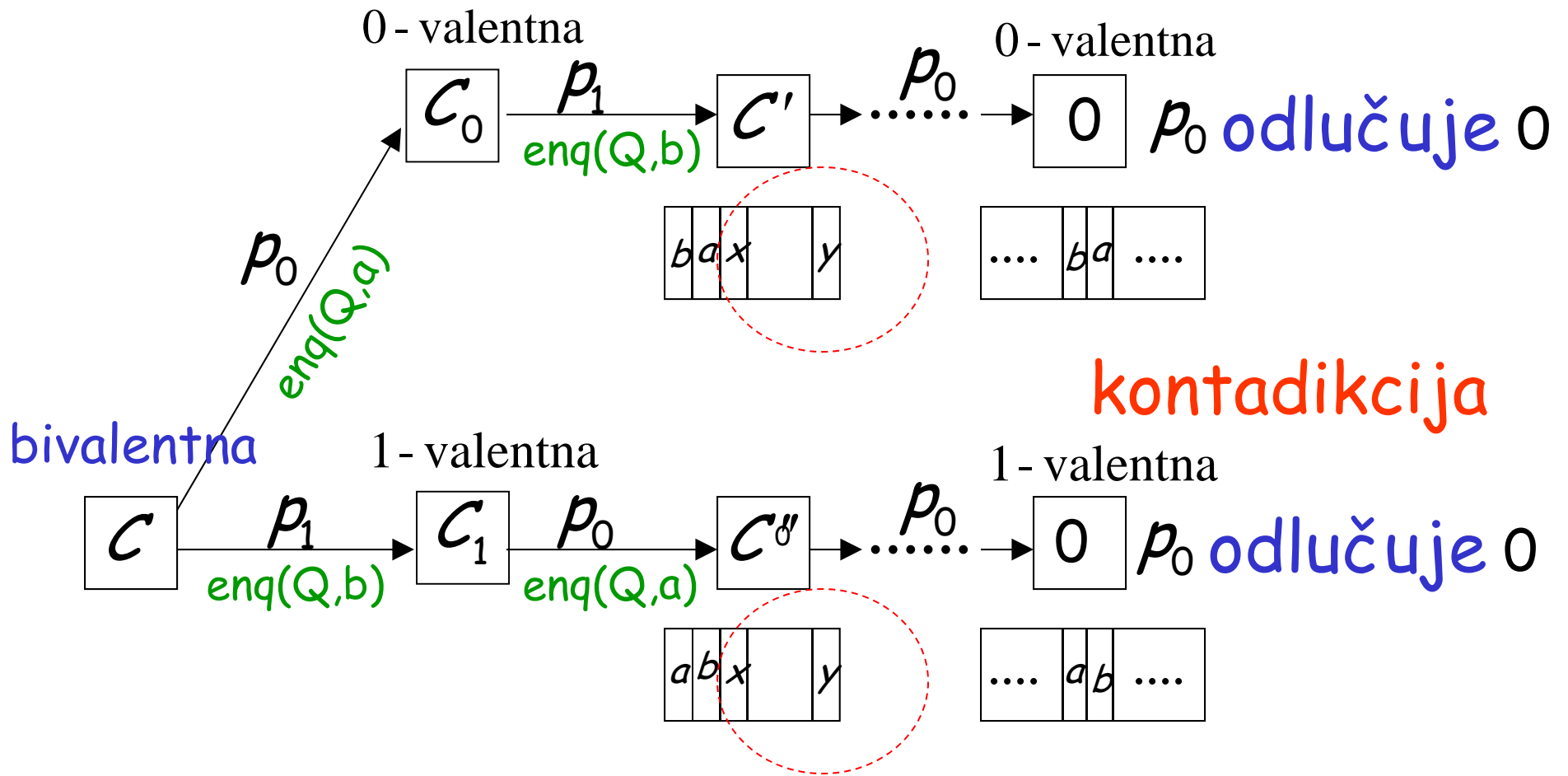


Objašnjenje



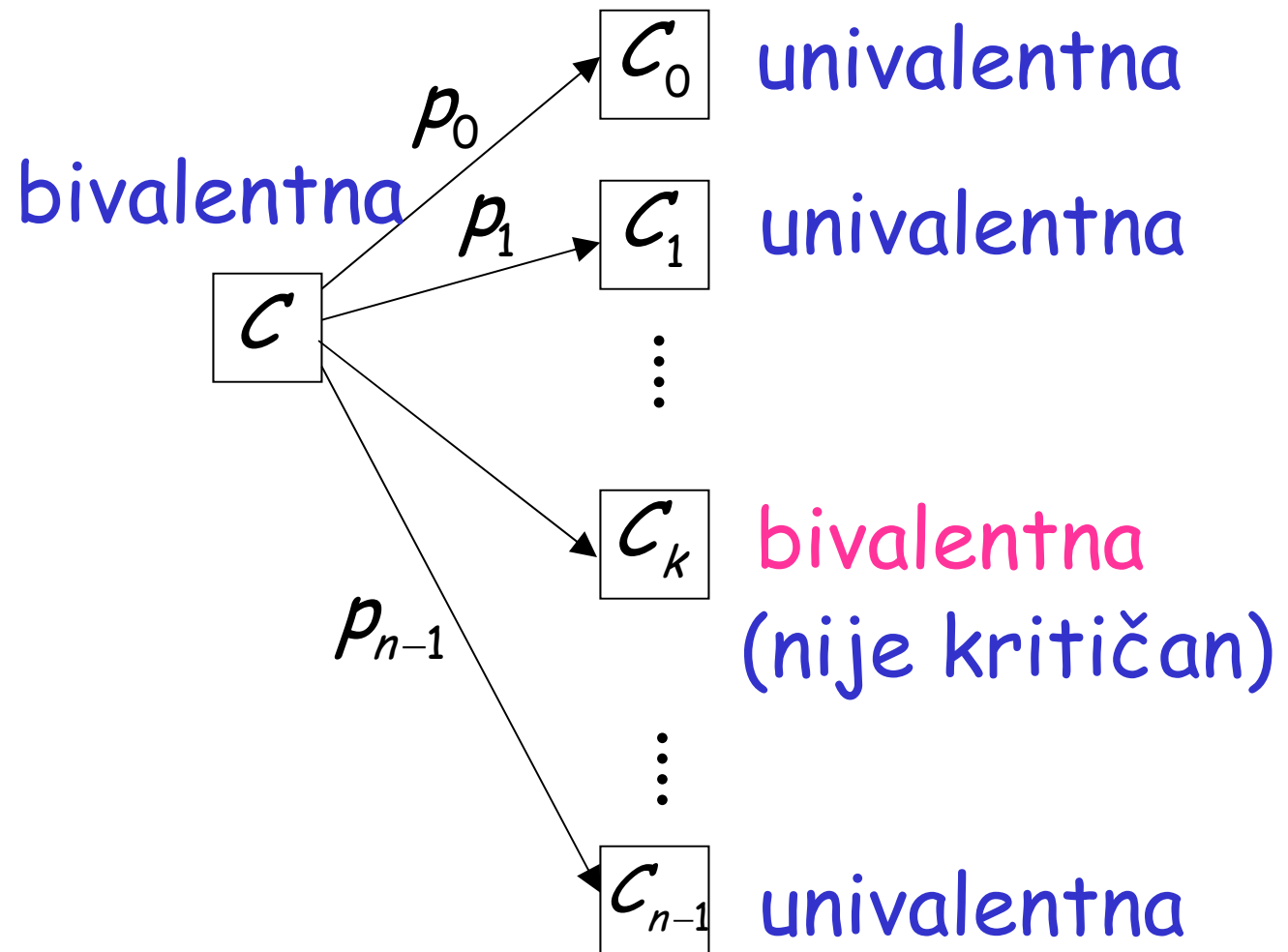
Predopst., p_0 ne radi dequeue a

Neka odluka će biti doneta pošto je
algoritam konsenzusa oslobođen-čekanja

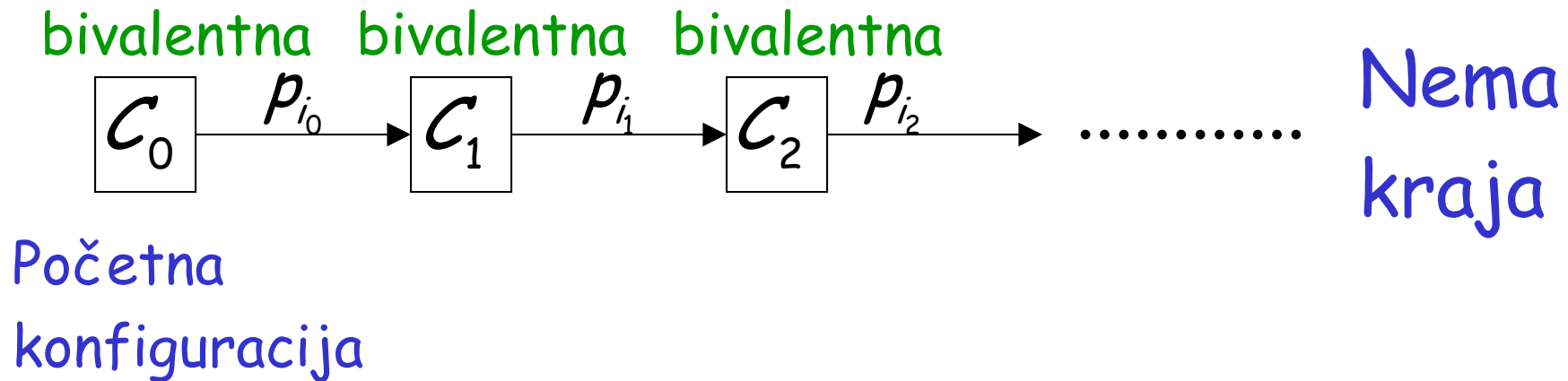


p_0 će se odlučiti za iste vrednosti,
pošto p_0 vidi iste vred. u deljenoj
memoriji u oba izvršenja

U svim slučajevima smo dobili kontradikciju;
Zbog toga, postoji procesor
koji nije kritičan



Zbog toga, možemo konstruisati izvršenje



Konsenzus se nikada ne može postići

Kraj dokaza teoreme

Compare&Swap

Compare&Swap(X,A,B)

{ Temp \leftarrow X;

If $X == A$ then $X \leftarrow B$;

Return Temp; }

Deljena memorija



X

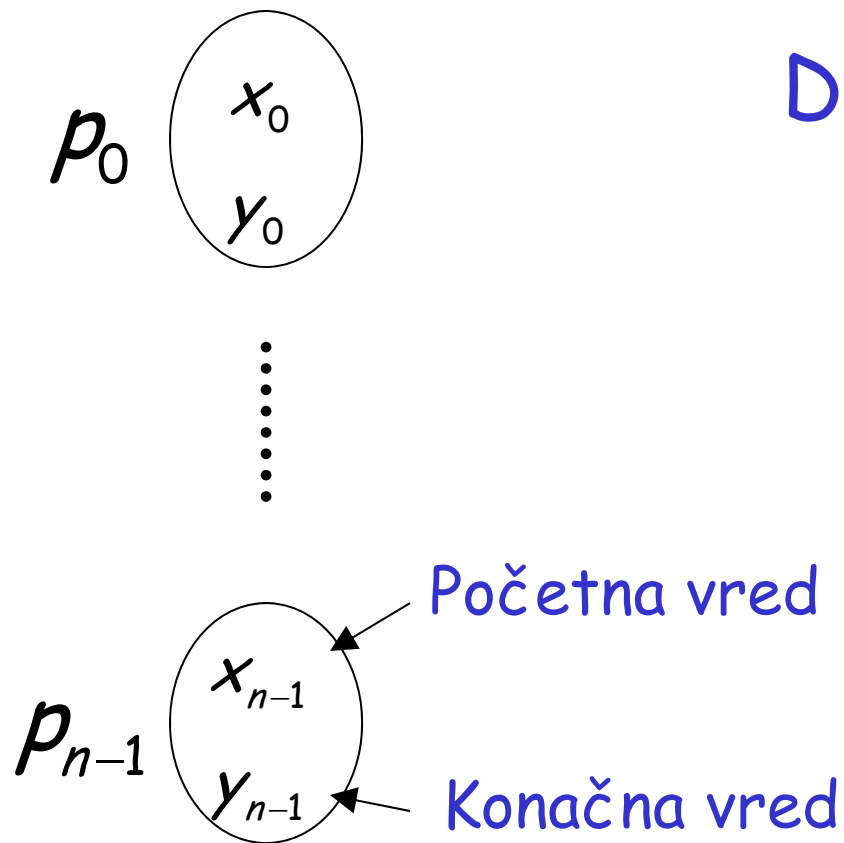
Teorema: Broj konsenzusa za
Compare&Swap objekt je ∞

Dokaz:

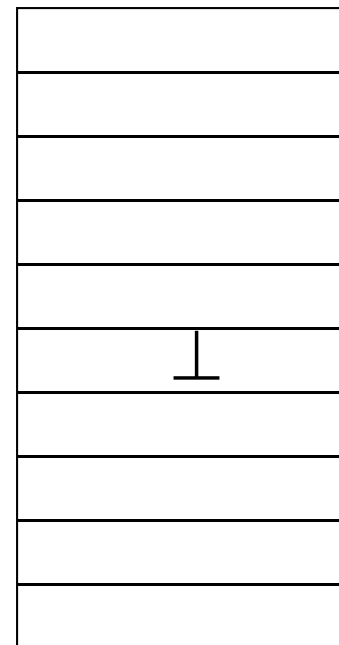
Sa n procesa, za bilo koje n ,
možemo rešiti WF konsenzus
koristeći Compare&Swap objekt
(i read/write objekte)

Jedan algoritam WF konsenzusa za n procesora sa korišćenjem C&S objekta

Lokalna memorija



Deljena memorija



First
(compare&swap
objekt)

Kod za procesor p_i $i \in \{0,1\}$

$v_i \leftarrow \text{Compare \& Swap}(\text{First}, \perp, x_i)$

If $v_i == \perp$ //da li sam ja prvi?

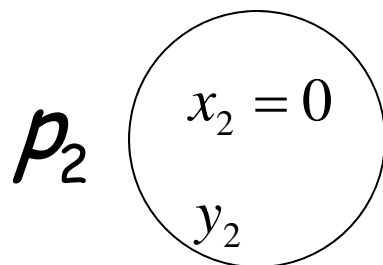
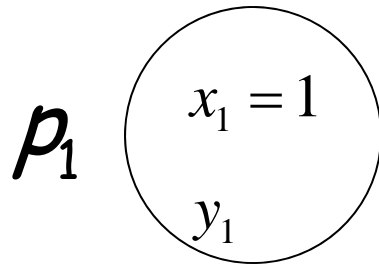
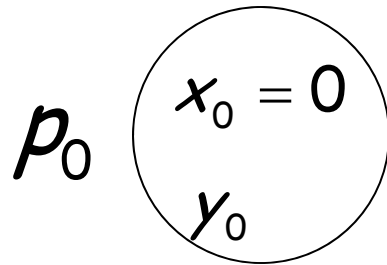
then $y_i \leftarrow x_i$ //da, izaberi moju vred

else $y_i \leftarrow v_i$ //ne, izaberi vrednost
od prvog procesa
koja je smeštena u First

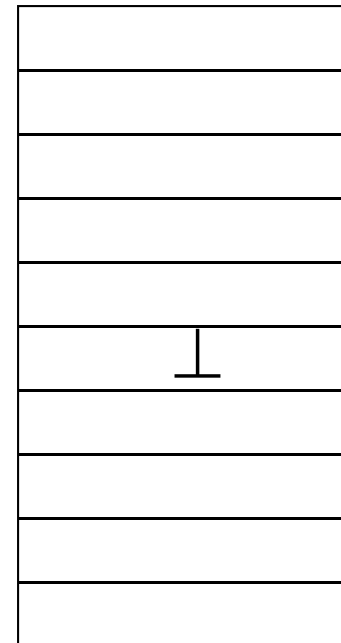
Napomena: ovaj algoritam koristi compare&swap
i read/write objekte

Primer izvršenja:

Lokalna memorija



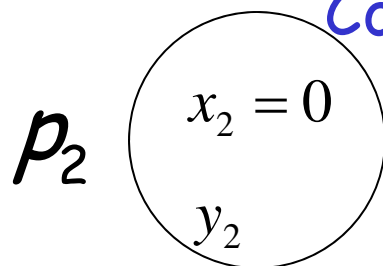
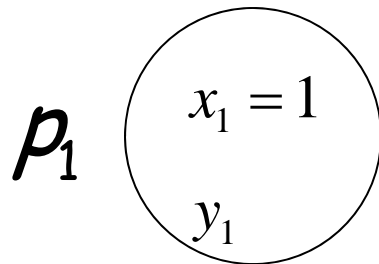
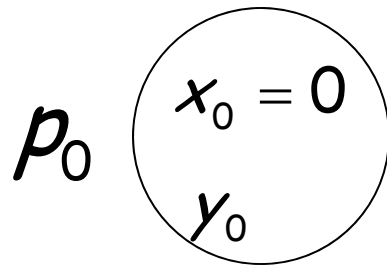
Deljena memorija



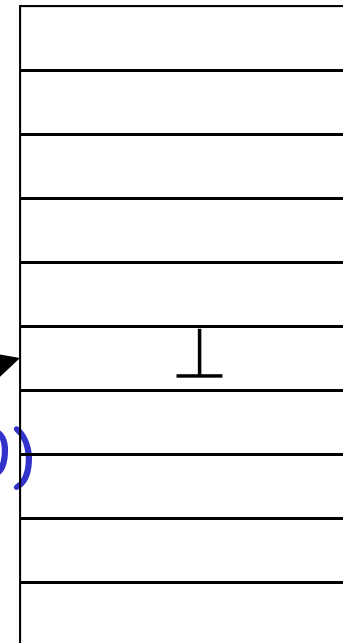
First

Predpost. da se p_2 izvršava prvi

Lokalna memorija



Deljena memorija

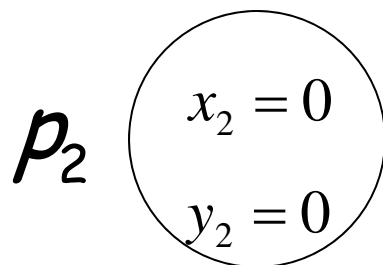
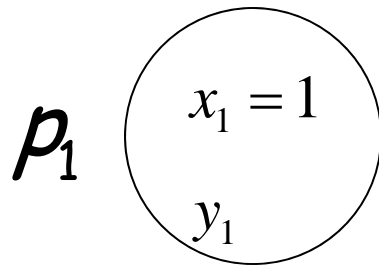
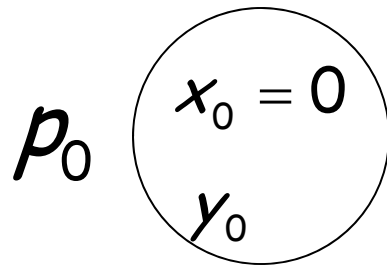


First

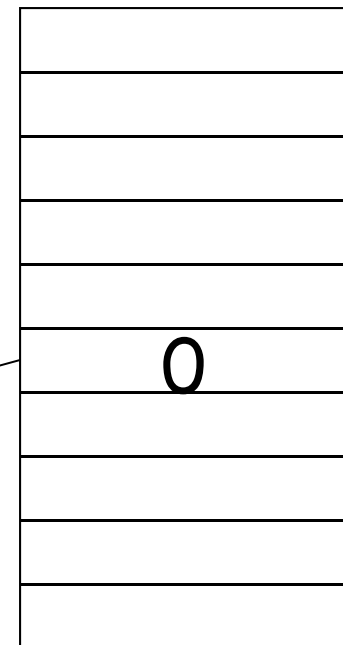
Compare&Swap(First, \perp , 0)

Predpost. da se p_2 izvršava prvi

Lokalna memorija



Deljena memorija

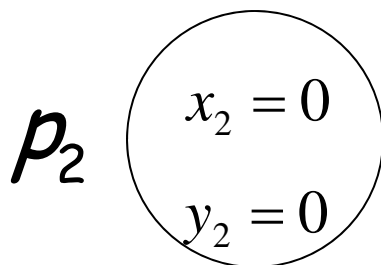
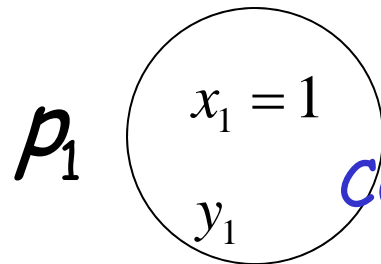
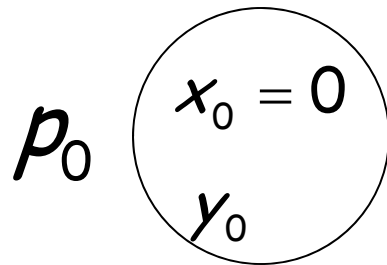


First

Shvata da je prvi, odlučuje se za svoju vred

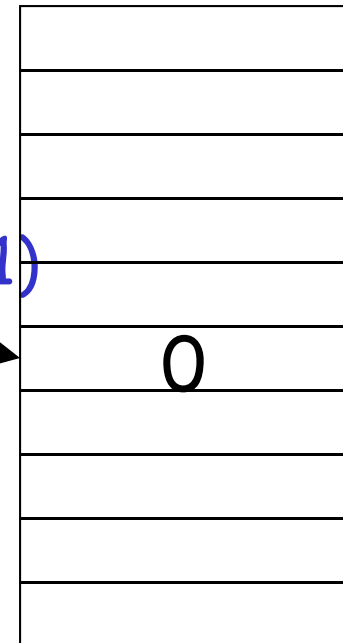
Predpost. da se p_1 izvršava drugi

Lokalna memorija



Deljena memorija

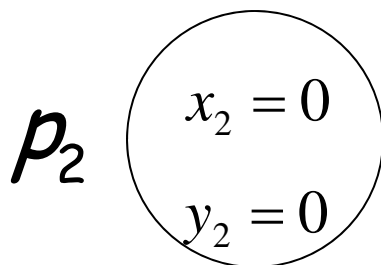
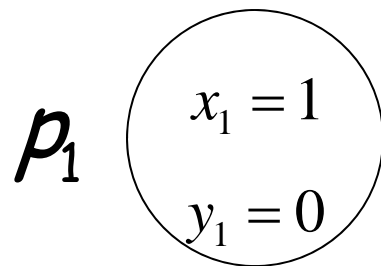
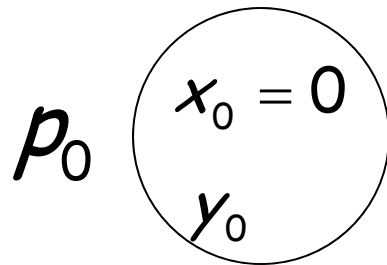
Compare&Swap(First, \perp , 1)



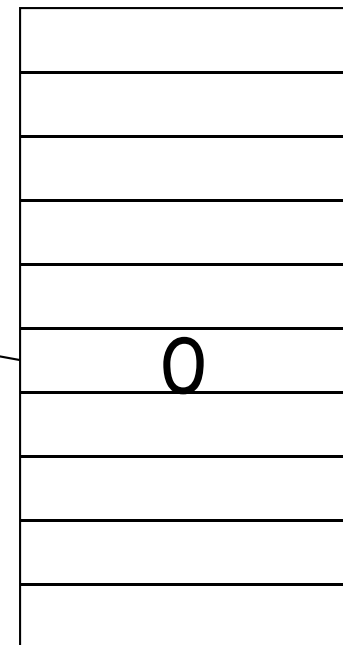
First

Predpost. da se p_1 izvršava drugi

Lokalna memorija



Deljena memorija



First

0

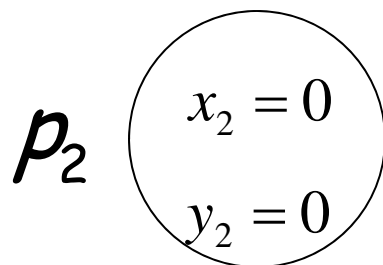
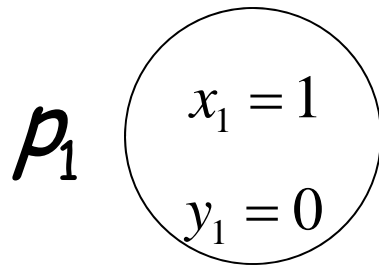
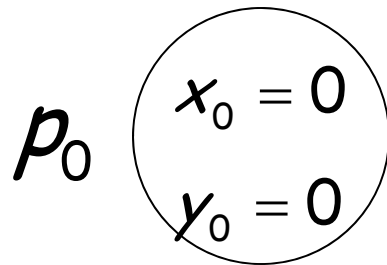
0

Shvata da nije prvi, odlučuje se za vred First⁴²

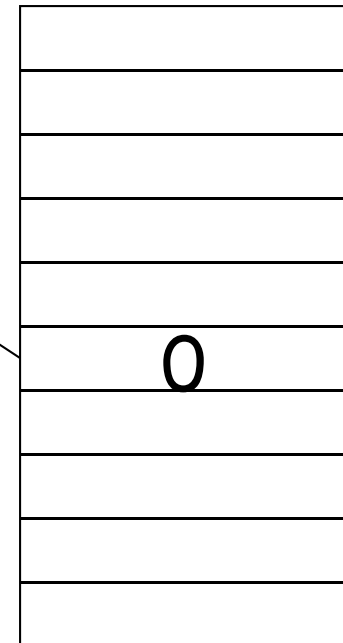
Slično za p_0

Konsenzus je
postignut

Lokalna memorija



Deljena memorija



First

0

0

Shvata da nije prvi, odlučuje se za vred First⁴³

Ovaj algoritam je WF, pošto posle završetka Compare&Swap operacije, svaki procesor odlučuje (bez razmatranja šta drugi procesori rade)

Hijerahija konsenzusa

Deo 3

Broj konsenzusa

Broj konsenzusa za objekt datog tipa:

Maksimalan broj procesa za koje
se objekt može koristiti za rešavanje
problema WF konsenzusa
(zajedno sa read/write objektima)

Tip objekta

Broj konsenzusa

Read/Write

1

FIFO

2

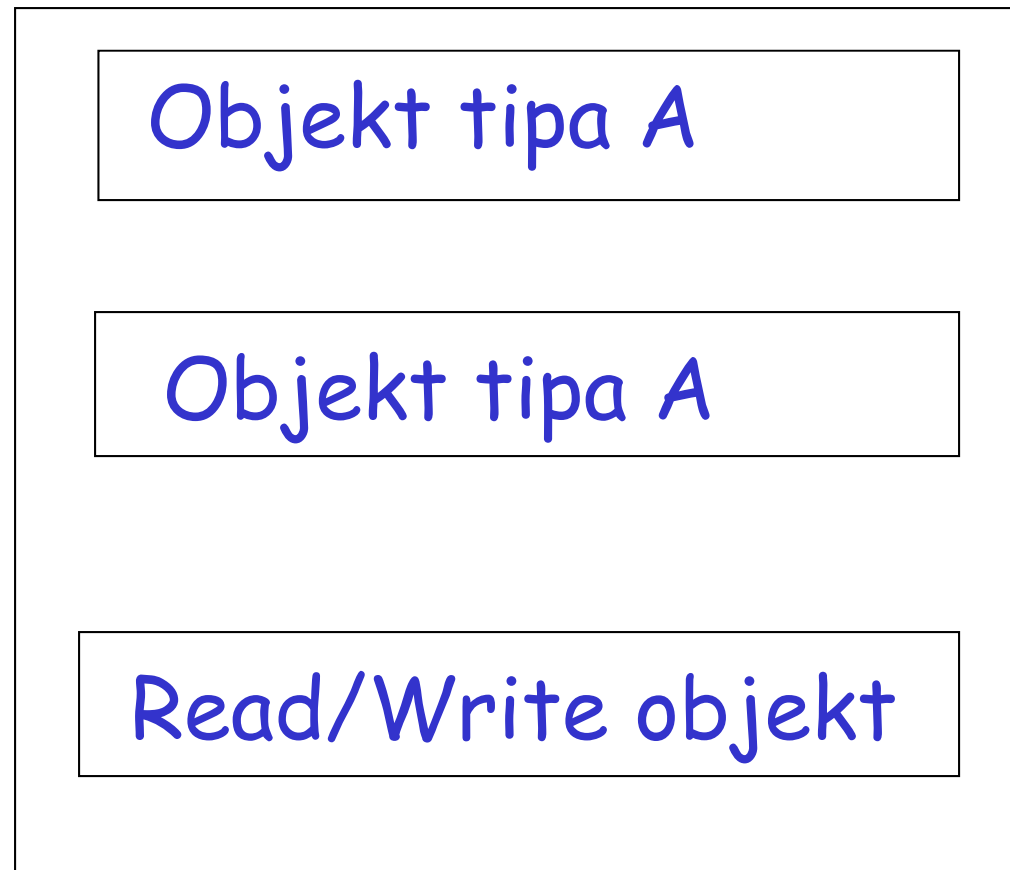
Compare&Swap

∞

(beskonačno)

Simulacija:

Objekt tipa B



Objekt tipa A simulira objekt tipa B

Teorema: Objekt tipa A sa
brojem konsenzusa n
ne može da simulira
drugi objekt tipa B sa
brojem konsenzusa $m > n$

Dokaz: Jer bi inače, objekt tipa A
imao broj konsenzusa m

Kraj dokaza

Univerzalnost

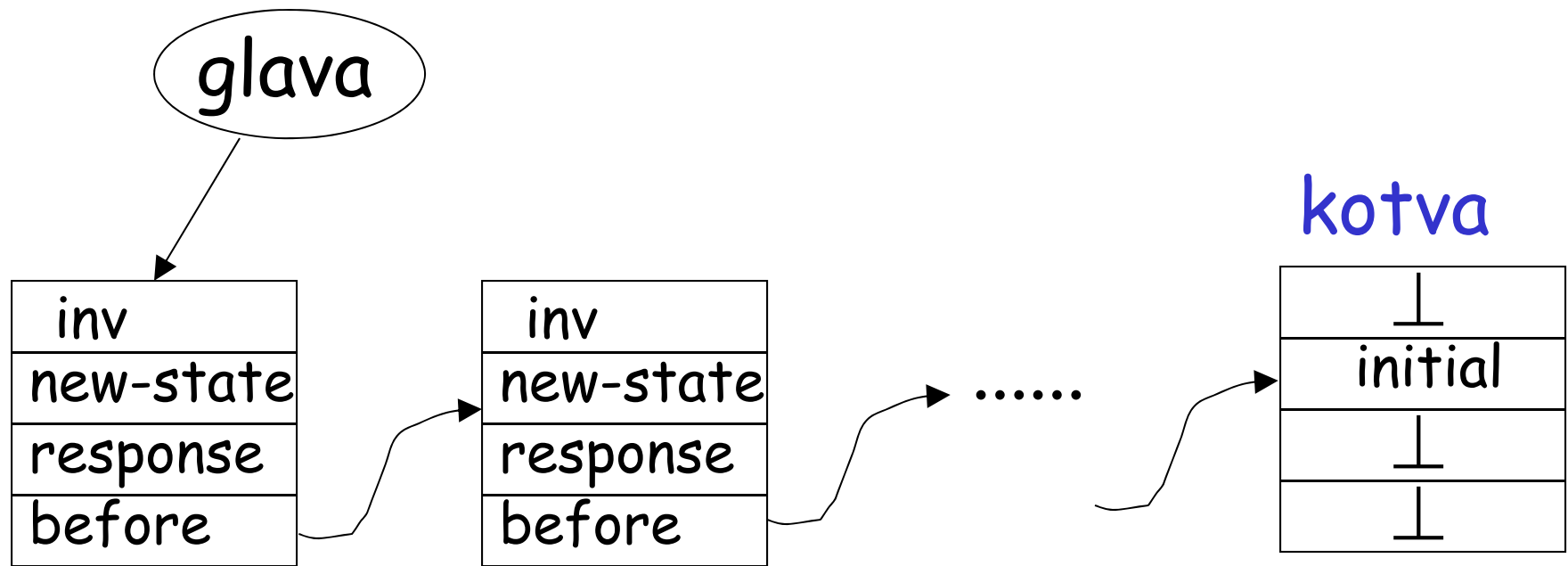
Univerzalni objekt:

može da simulira na WF način
bilo koji proizvoljni objekt

Možemo da pokažemo:

Objekti sa brojem konsenzusa n
mogu da simuliraju na WF način
bilo koji proizvoljan objekt za do n
procesora

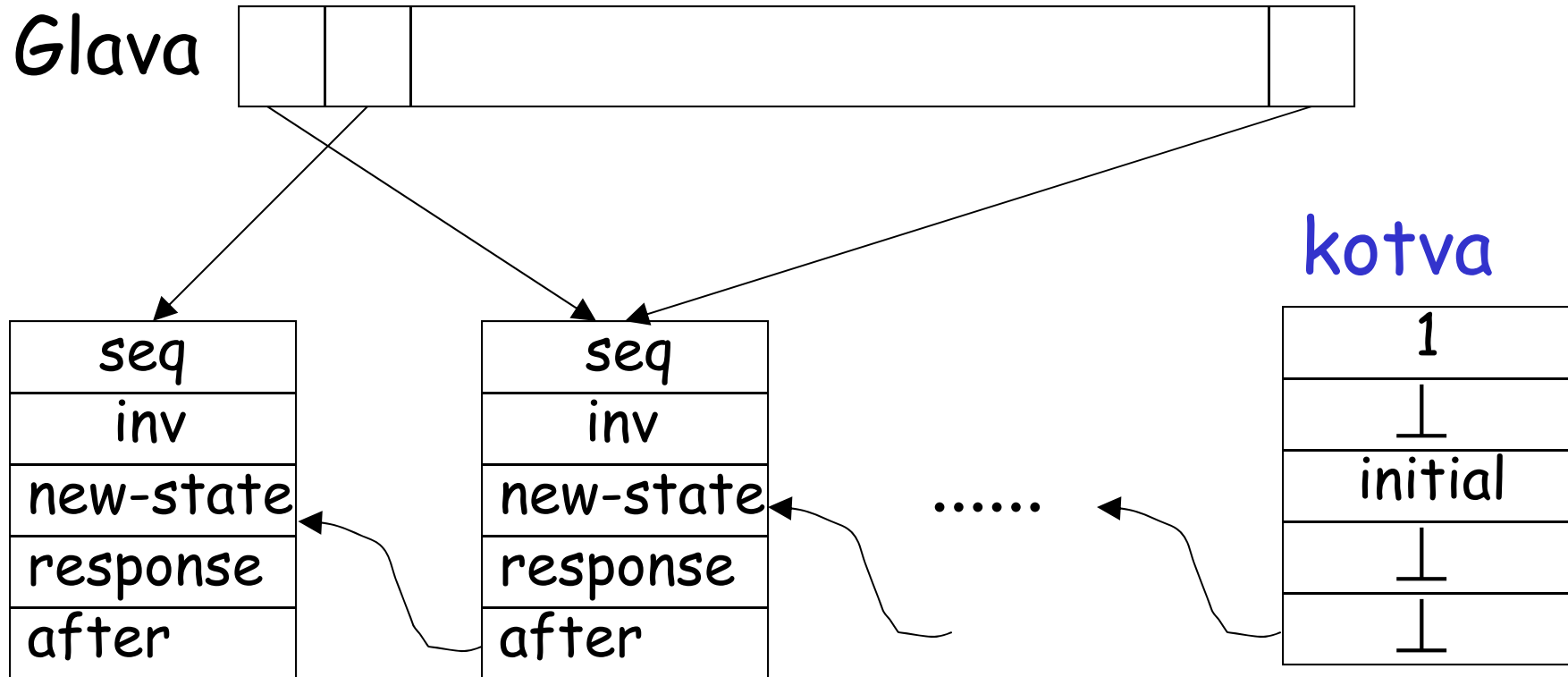
Compare&Swap



Simulacija bez blokiranja (Non-Blocking)

Simulacija bez blokiranja (Non-Blocking)

Glava[i] je pokazivač za i-ti proces



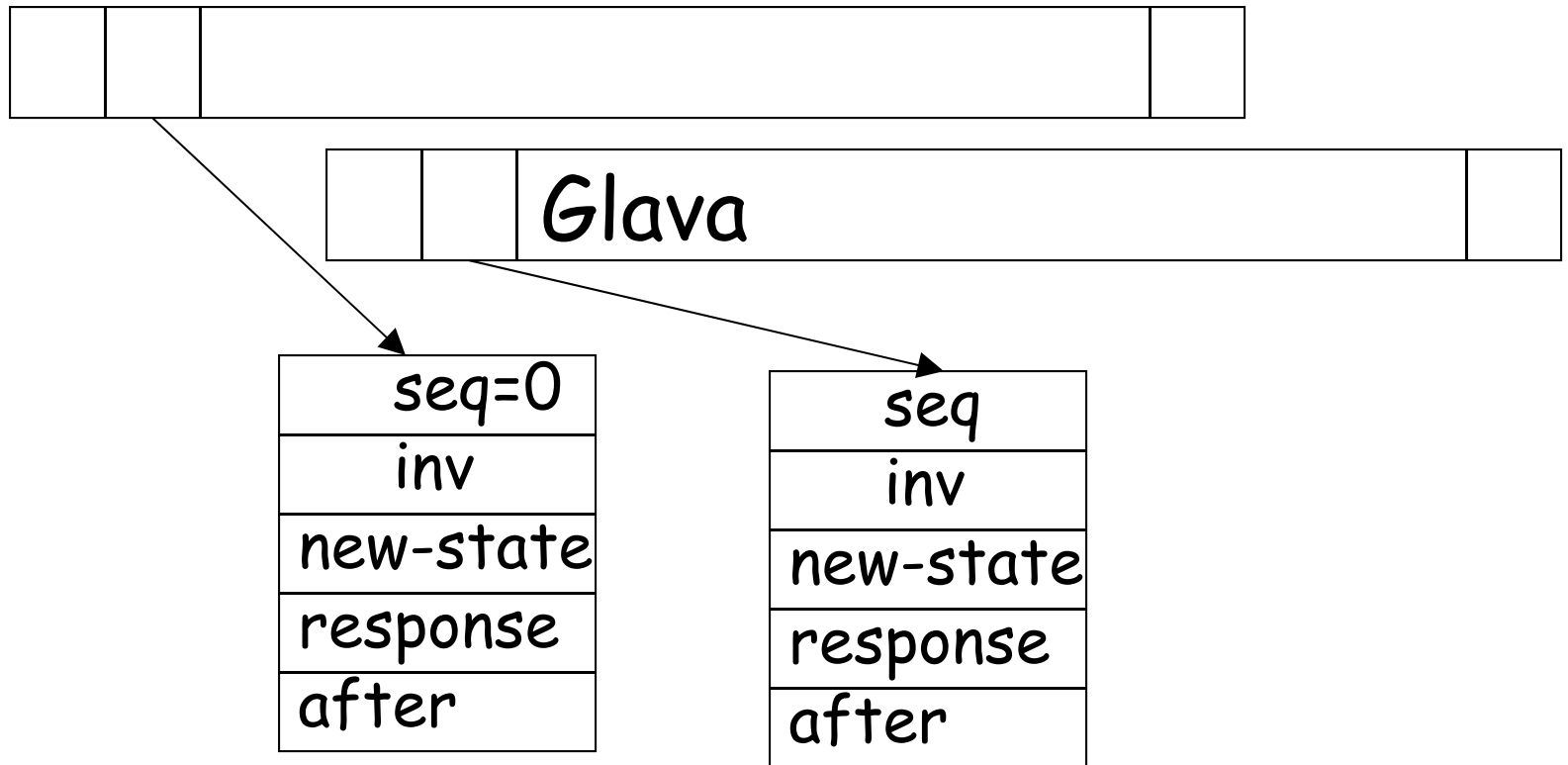
Polje after je objekt konsenzusa sa $CN = n$

Procesi se bore da se ulančaju na vrh liste

Simulacija oslobođena čekanja (Wait-free)

Najava[i] je pokazivač za i-ti proces

Najava



Prioritet ima proces p_i , $i = (\text{seqmax} + 1) \% n$

Ako čeka, svi mu pomažu, inače pomažu sebi