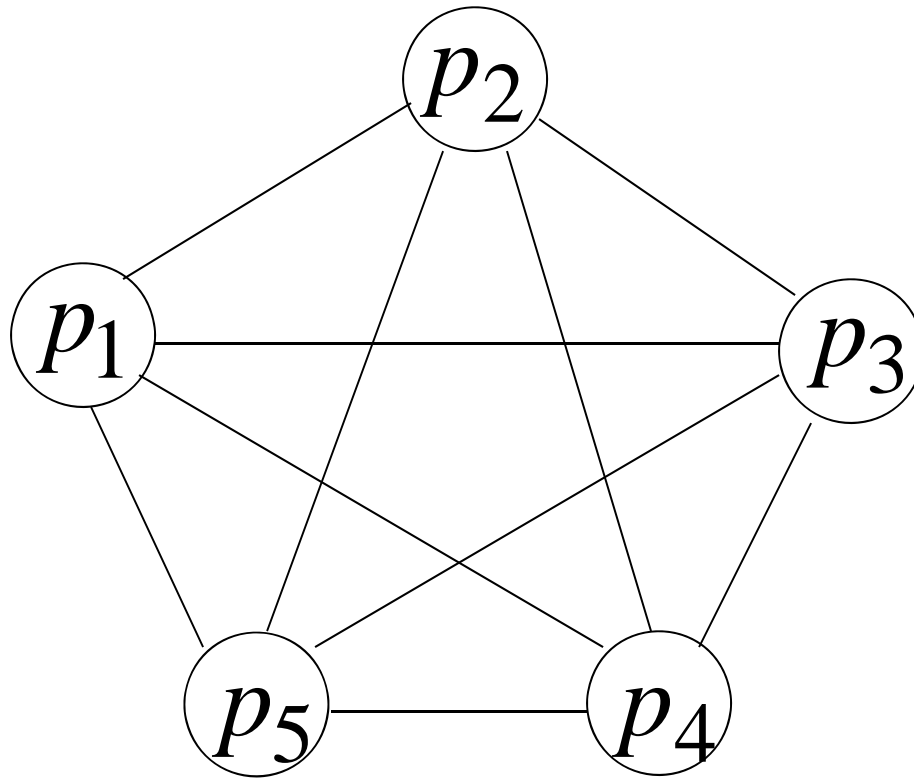


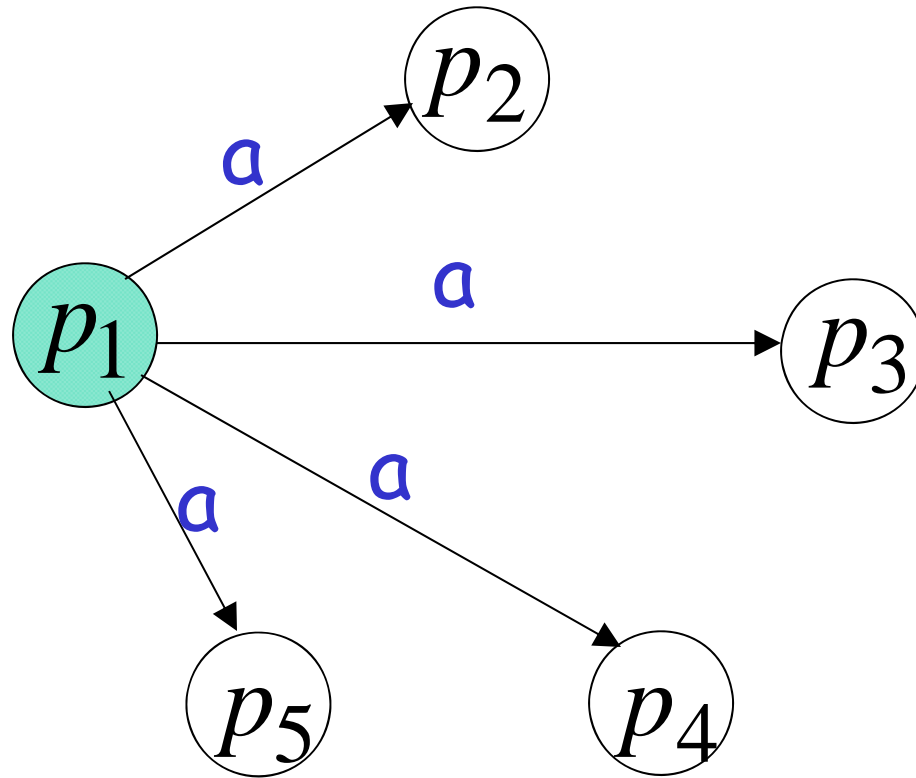
Konsenzus otporan na otkaze (Fault-Tolerant Consensus)

Model komunikacije

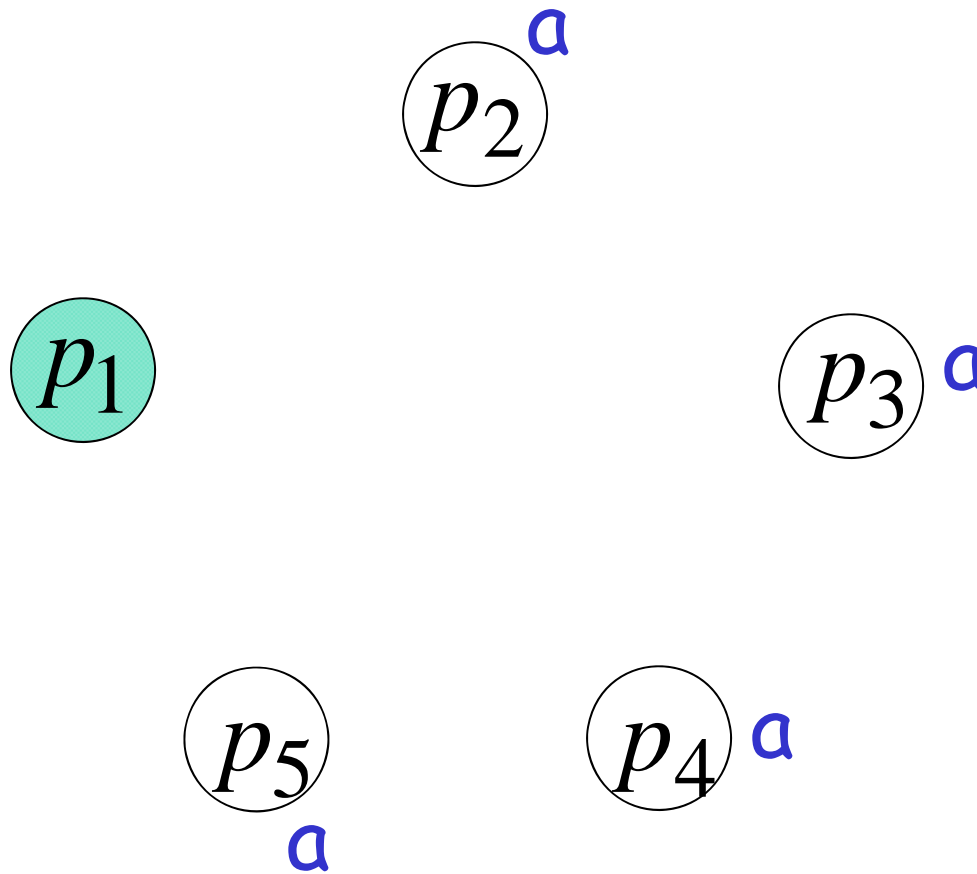


- Potpun graf (svaki čvor povezan sa svakim)
- Sinhrona mreža

Slanje svima (Broadcast)

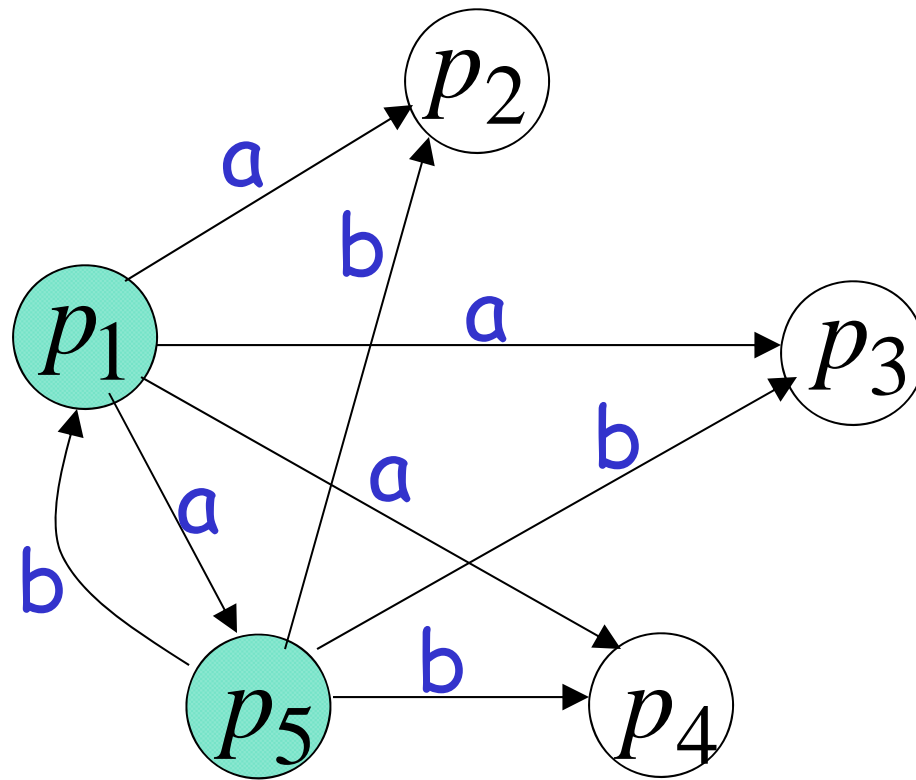


Slanje poruke a svim proc u jednoj rundi

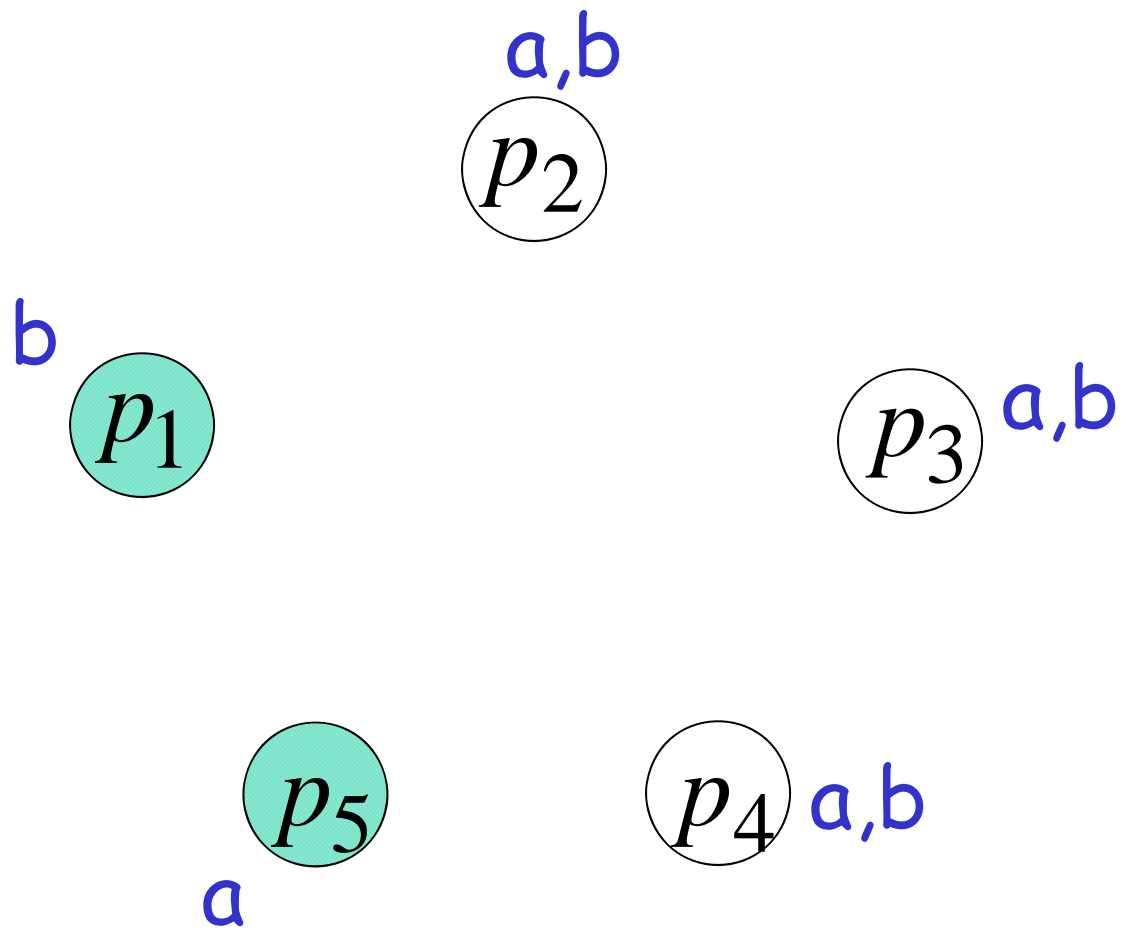


Na kraju runde: svi su primili a

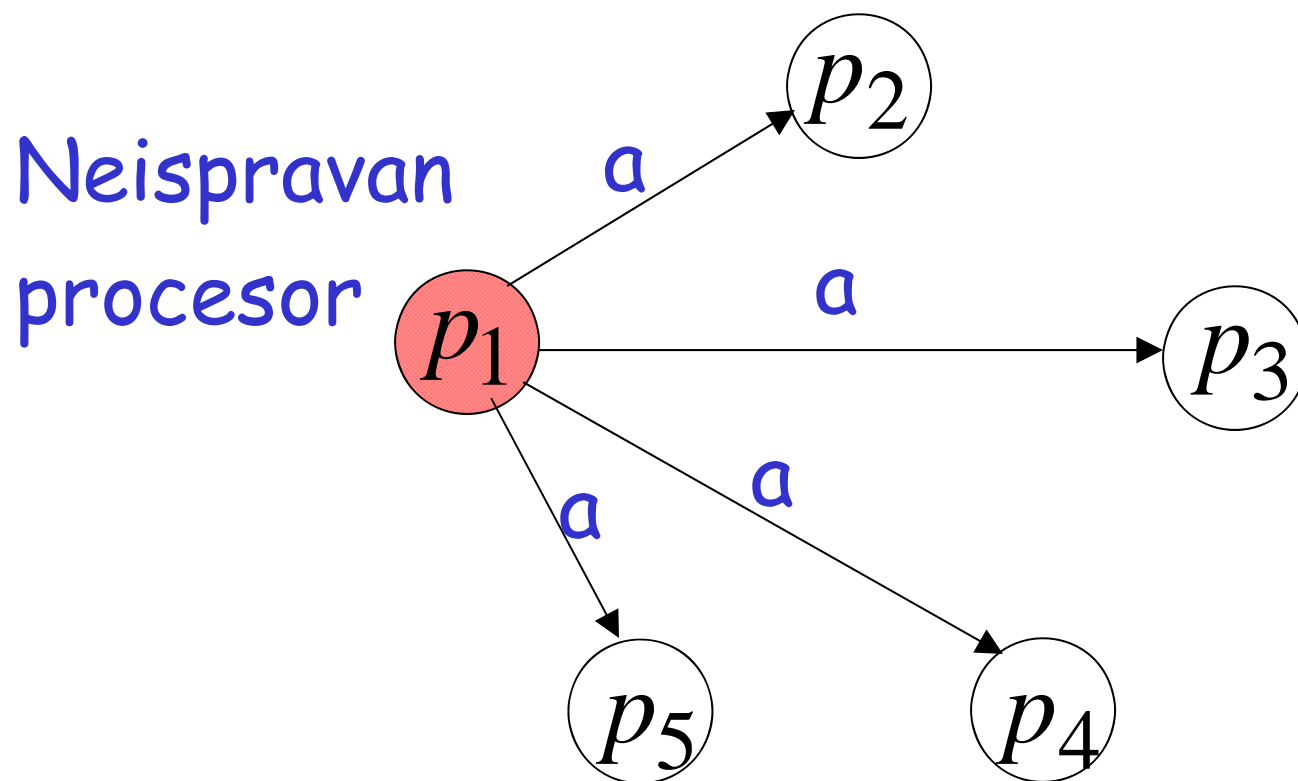
Slanje svima

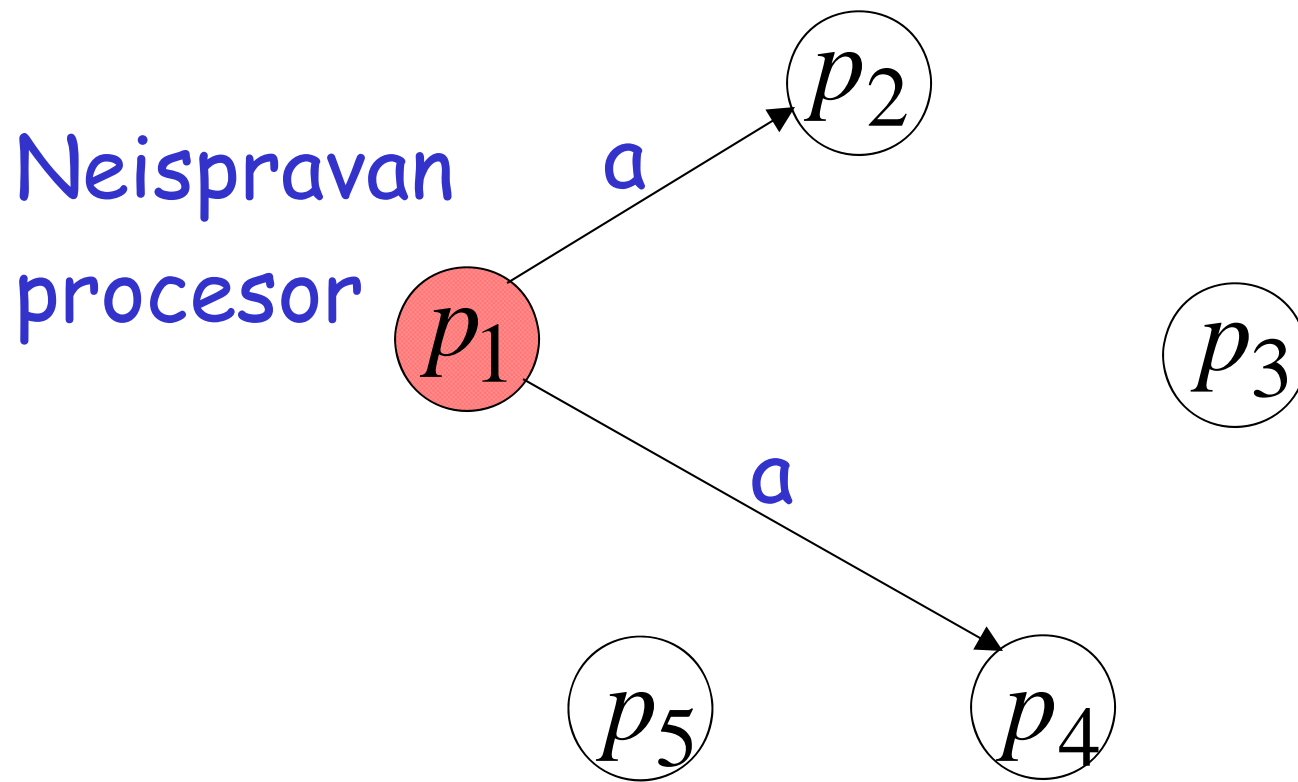


Dva ili više procesora mogu slati svima u istoj rudi



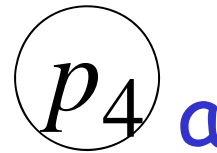
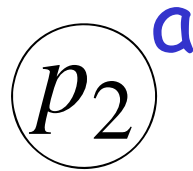
Otkazi tipa ispada (Crash)

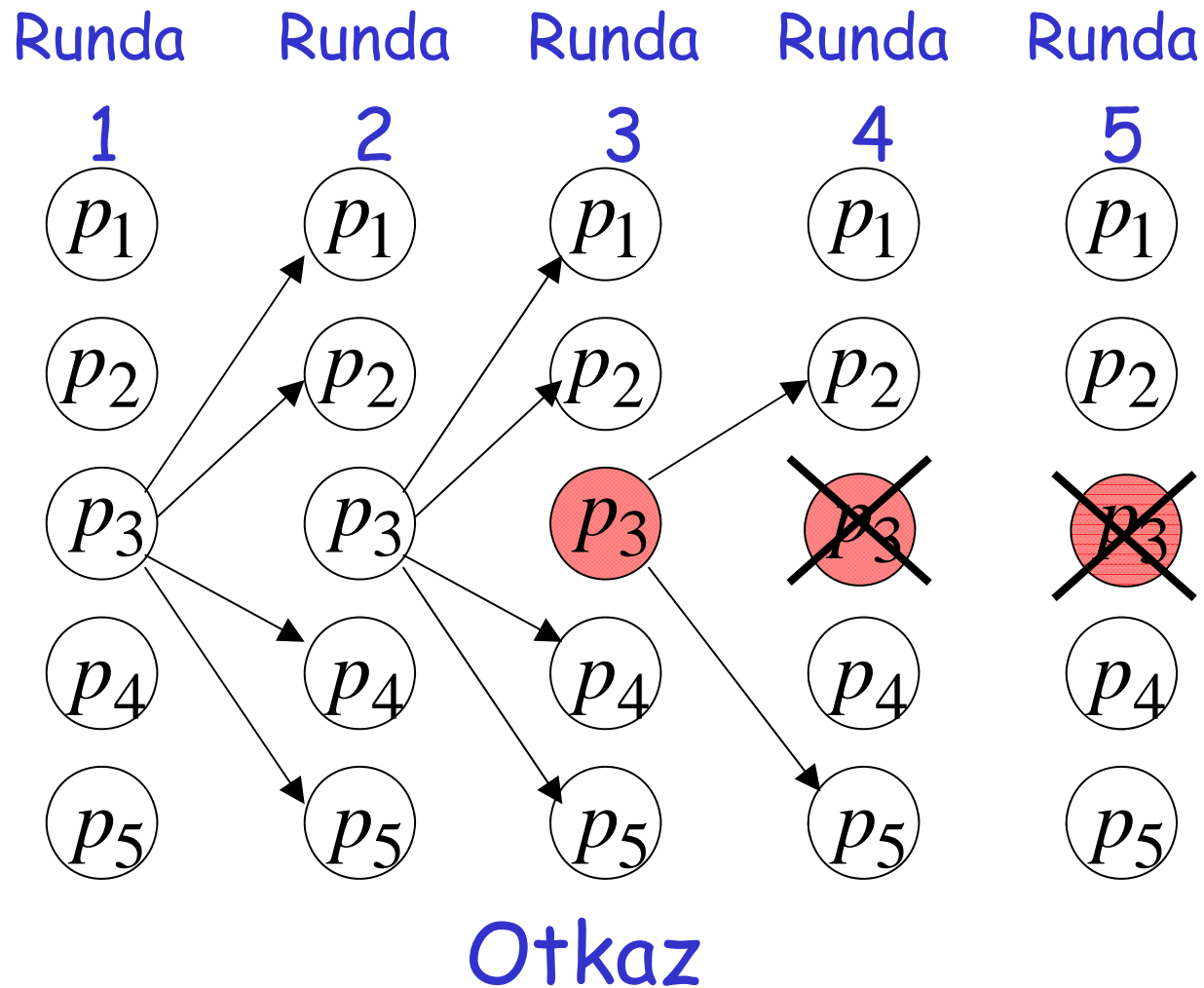




Neke od poruka se gube,
one nikad neће biti primljene

Neispravan
procesor

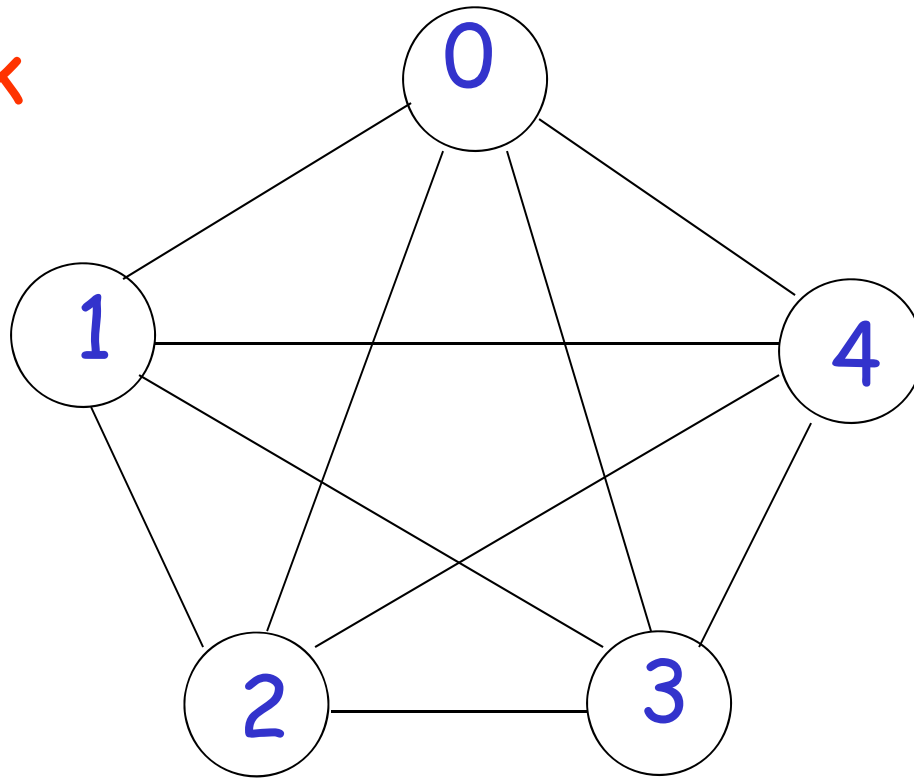




Posle otkaza proces nestaje iz mreže

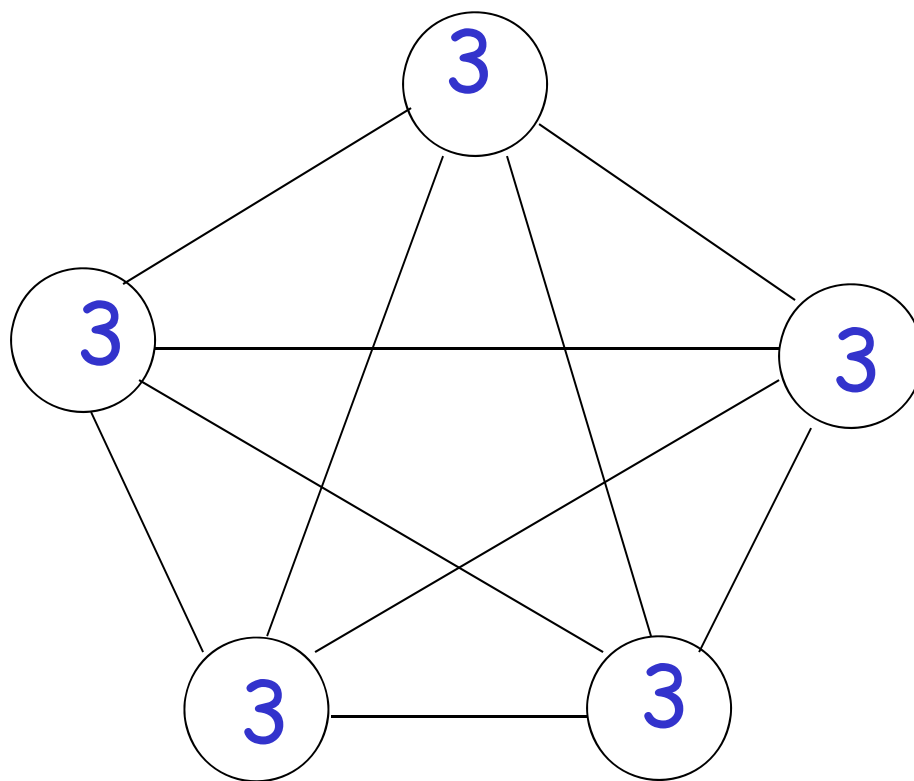
Konsenzus

Početak



Svako ima neku početnu vrednost

Kraj

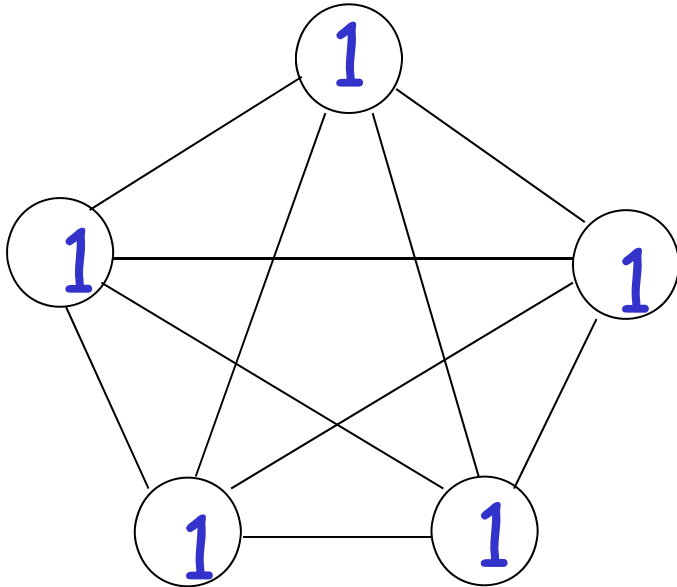


Svi moraju da se odluče za istu vrednost

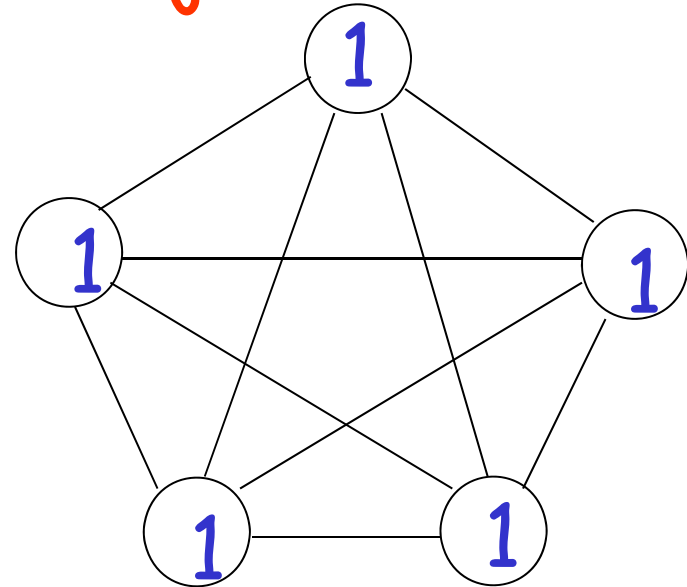
Uslov validnosti:

Ako svi počnu sa istom vrednošću,
oni moraju da se odluče za tu vrednost

Početak



Kraj



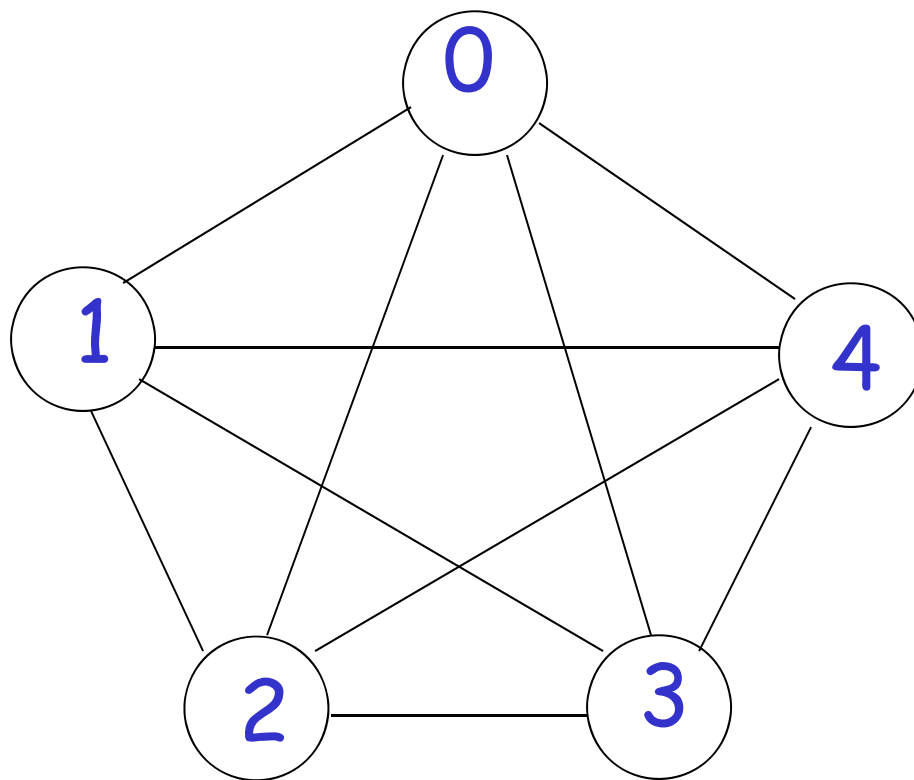
Jedan jednostavan algoritam

Svaki procesor:

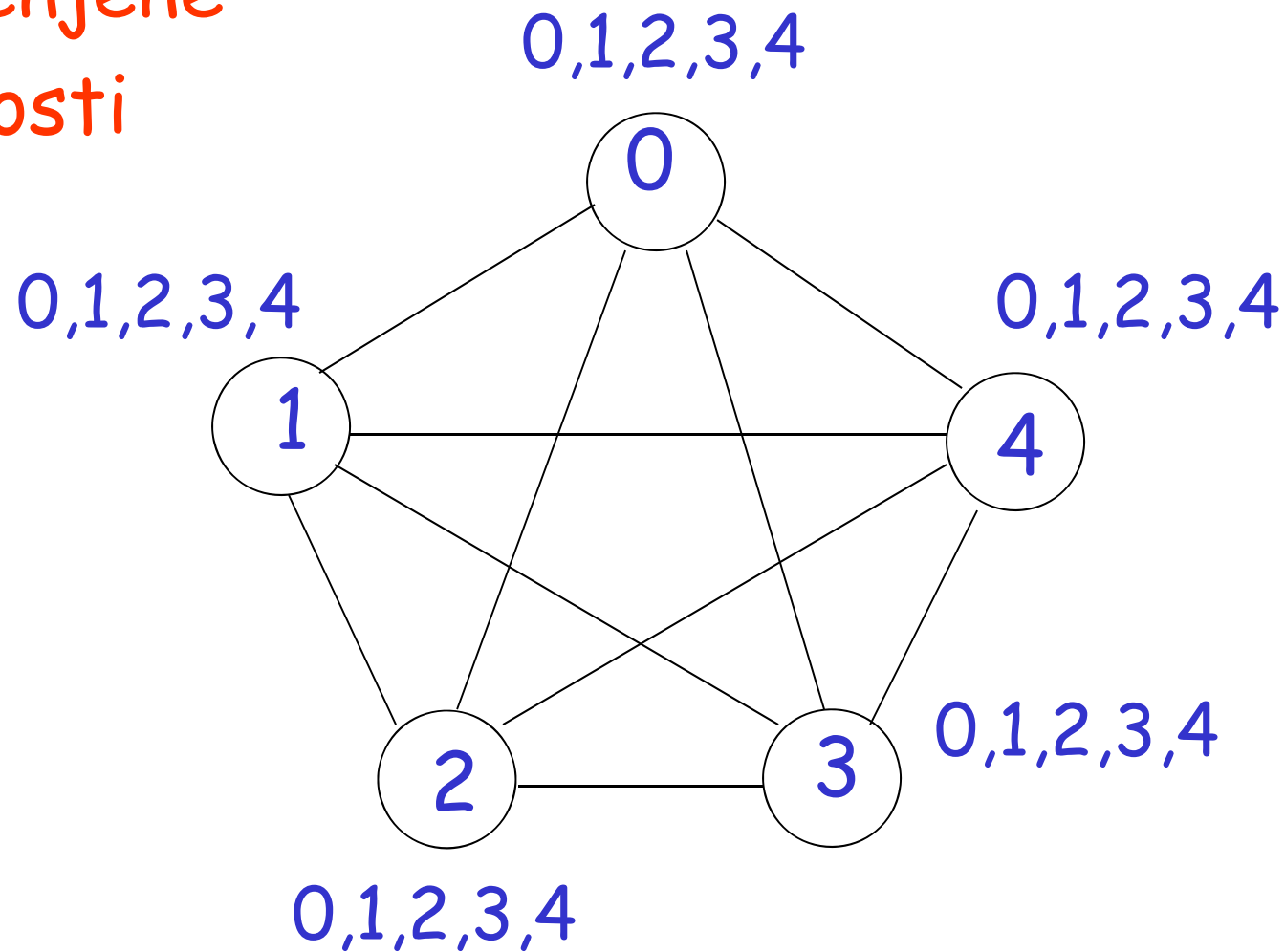
1. Pošalji vrednost svim procesorima
2. Odluči se za minimum

(potrebna je samo jedna runda)

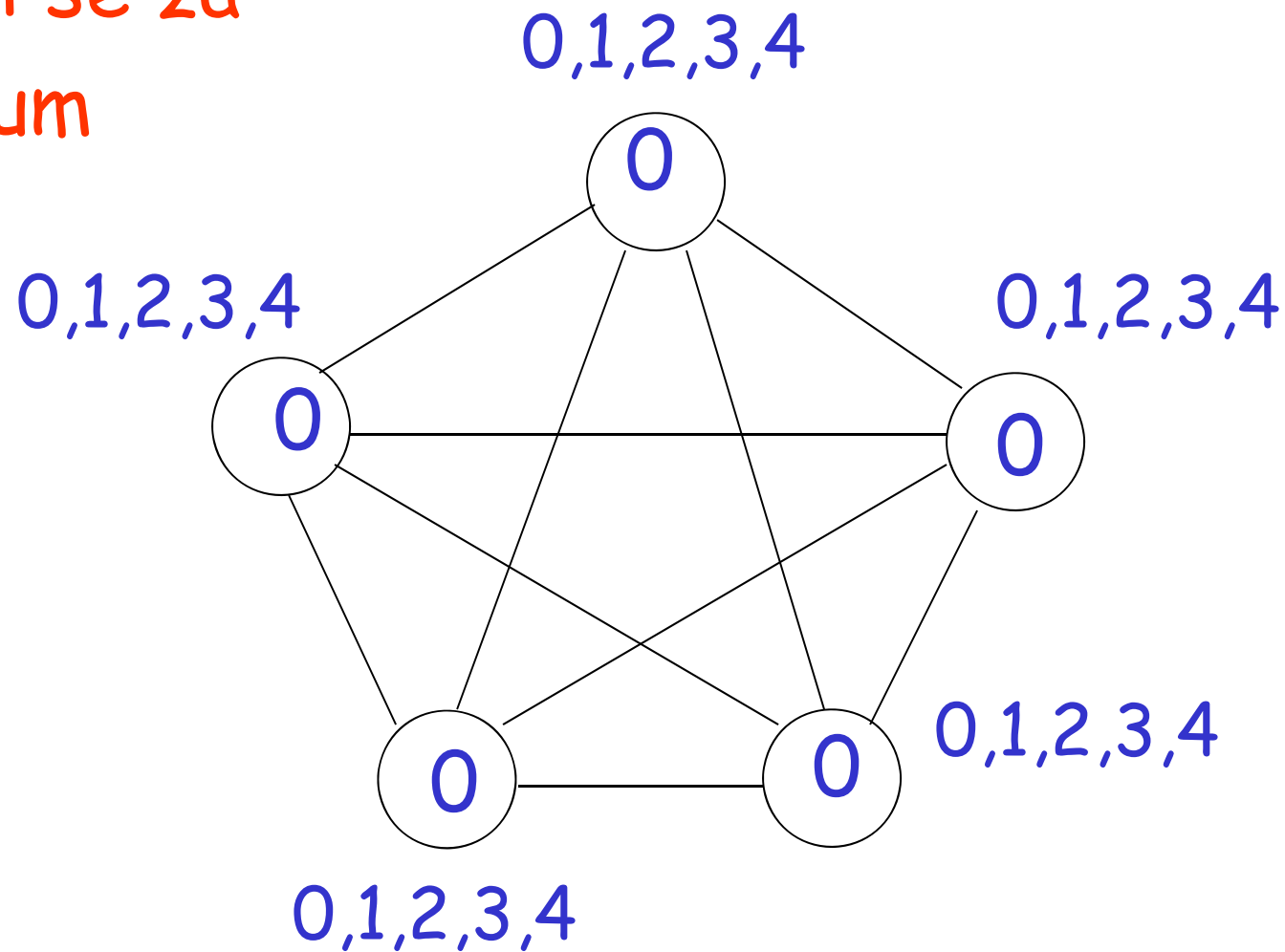
Početak



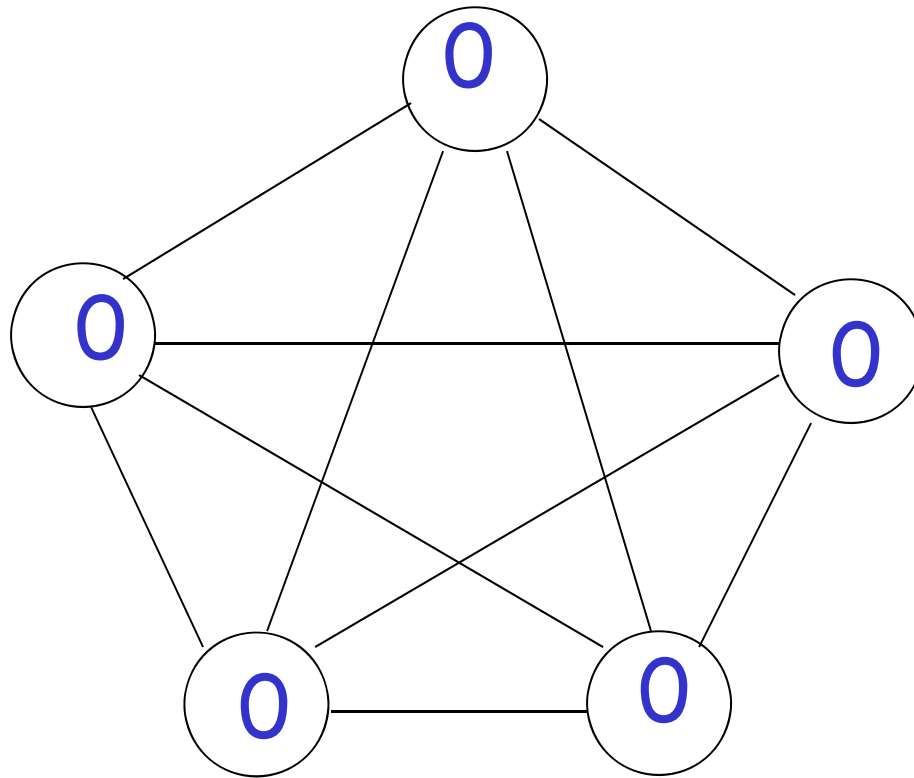
Razmenjene
vrednosti



Odluči se za
minimum

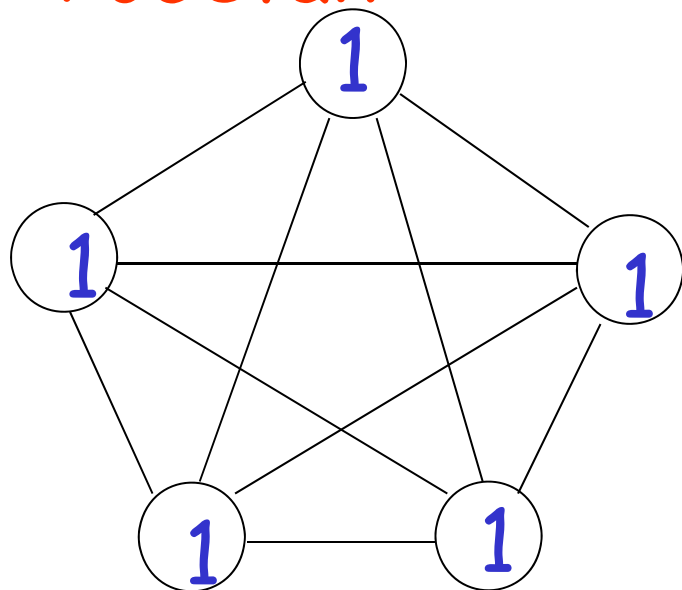


Kraj

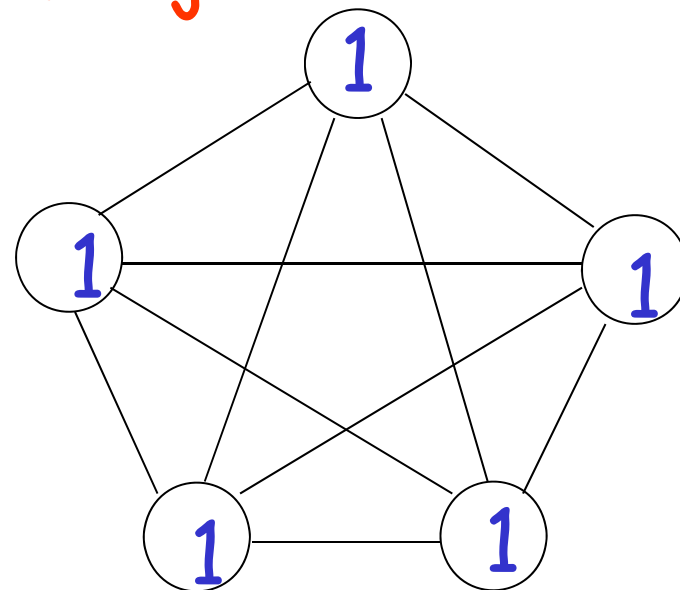


Ovaj algoritam zadovoljava uslov validnosti

Početak



Kraj



ako svi počnu sa istom početnom vrednošću,
svi se odlučuju za tu vrednost (minimum)

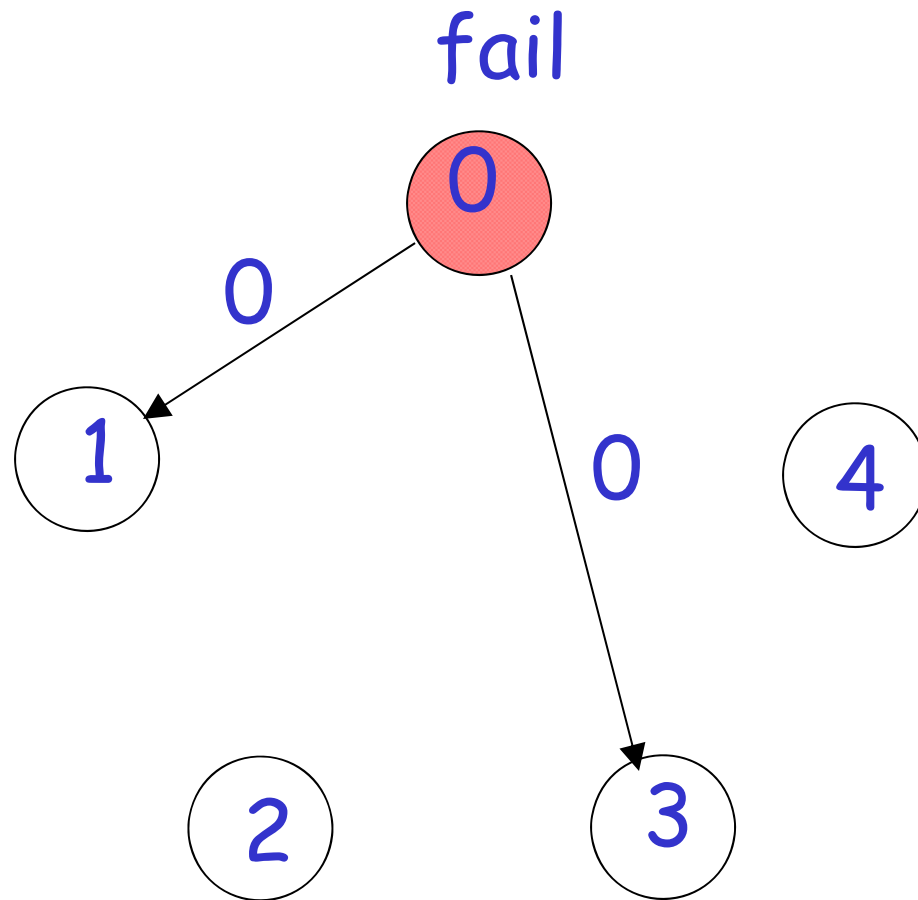
Konsenzus sa otkazima ispada

Ovaj jednostavan algoritam ne radi

Svaki procesor:

1. Šalje vrednost svim procesorima
2. Odlučuje se za minimum

Početak



Neispravan procesor ne šalje
svoju vrednost svim procesorima

Razmenjene vrednosti

ispao

0

0,1,2,3,4

1

1,2,3,4

4

1,2,3,4

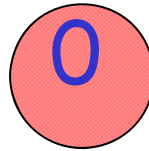
2

0,1,2,3,4

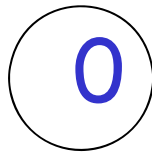
3

Odluči se za minimum

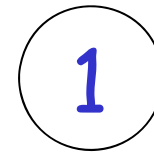
ispao



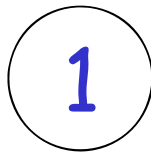
0,1,2,3,4



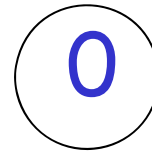
1,2,3,4



1,2,3,4

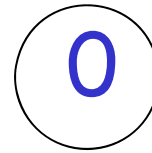
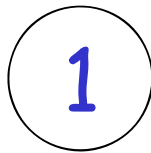
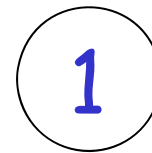
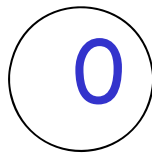
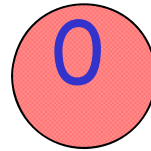


0,1,2,3,4



Kraj

ispao



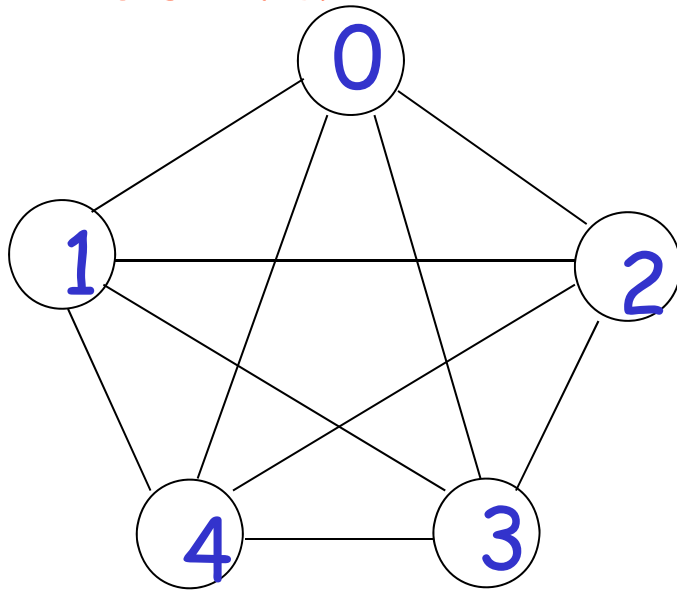
Nema konsenzusa!!!

Ako neki algoritam rešava konsenzus za f procesa u otkazu, kažemo da je on:

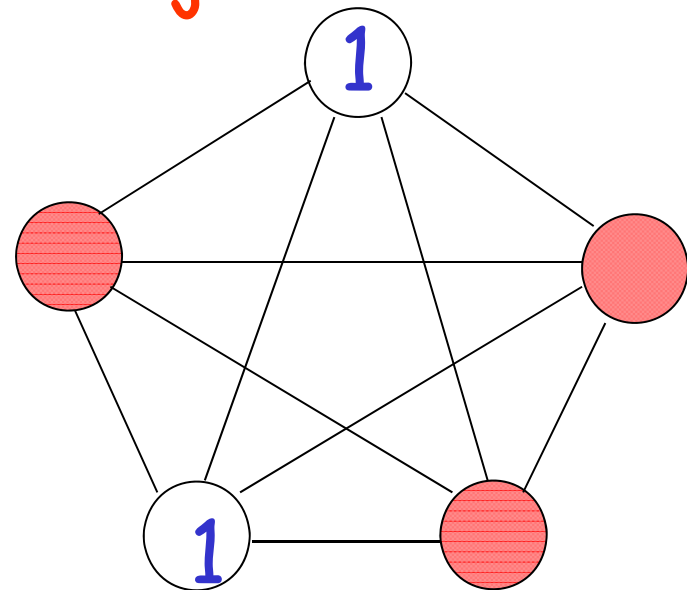
f -elastičan algoritam konsenzusa

Primer: Ulaz i izlaz jednog
3-elastičnog algoritma konsenzusa

Početak



Kraj



Jedan f-elastičan algoritam

Runda 1:

pošalji svima svoju vrednost

Runda 2 do runde $f+1$:

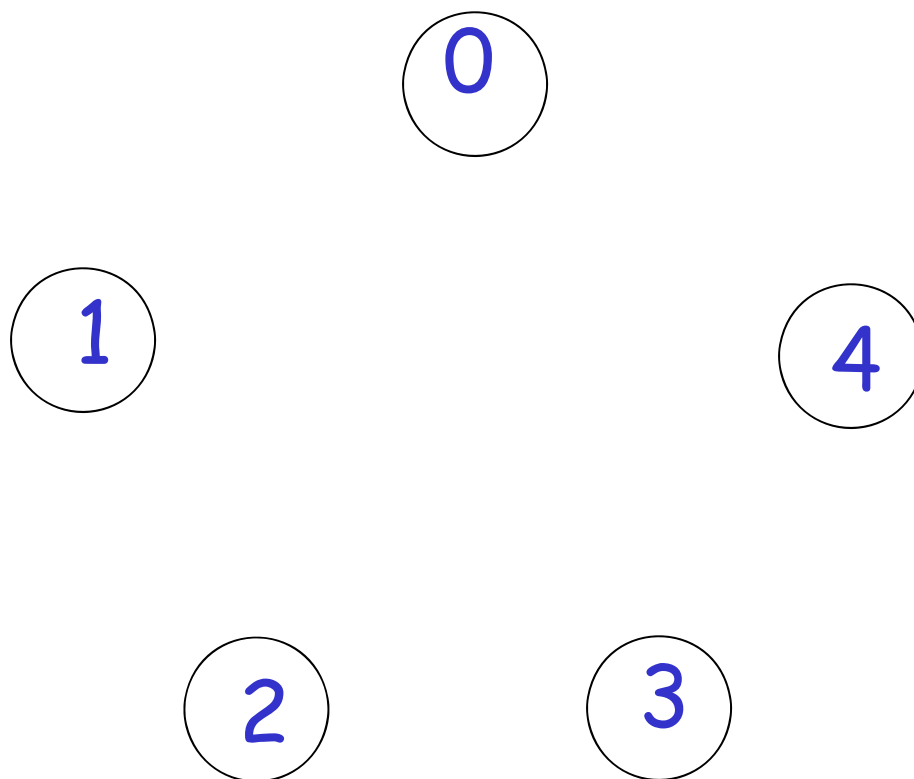
pošalji svima sve novo primljene vred.

Kraj runde $f+1$:

odluči se za min. primljenu vrednost

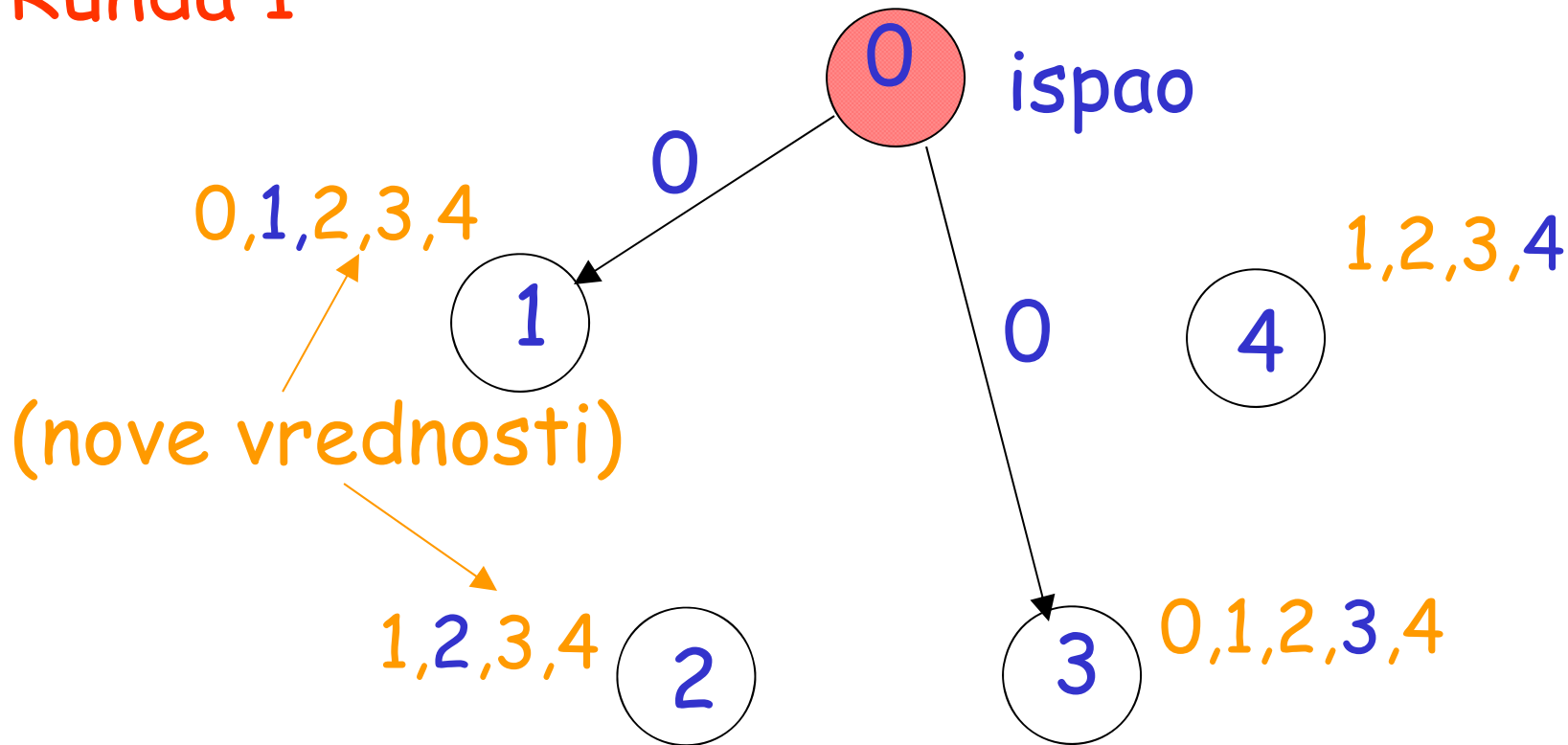
Primer: $f=1$ otkaza, $f+1 = 2$ potrebne runde

Početak



Primer: $f=1$ otkaza, $f+1 = 2$ potrebne runde

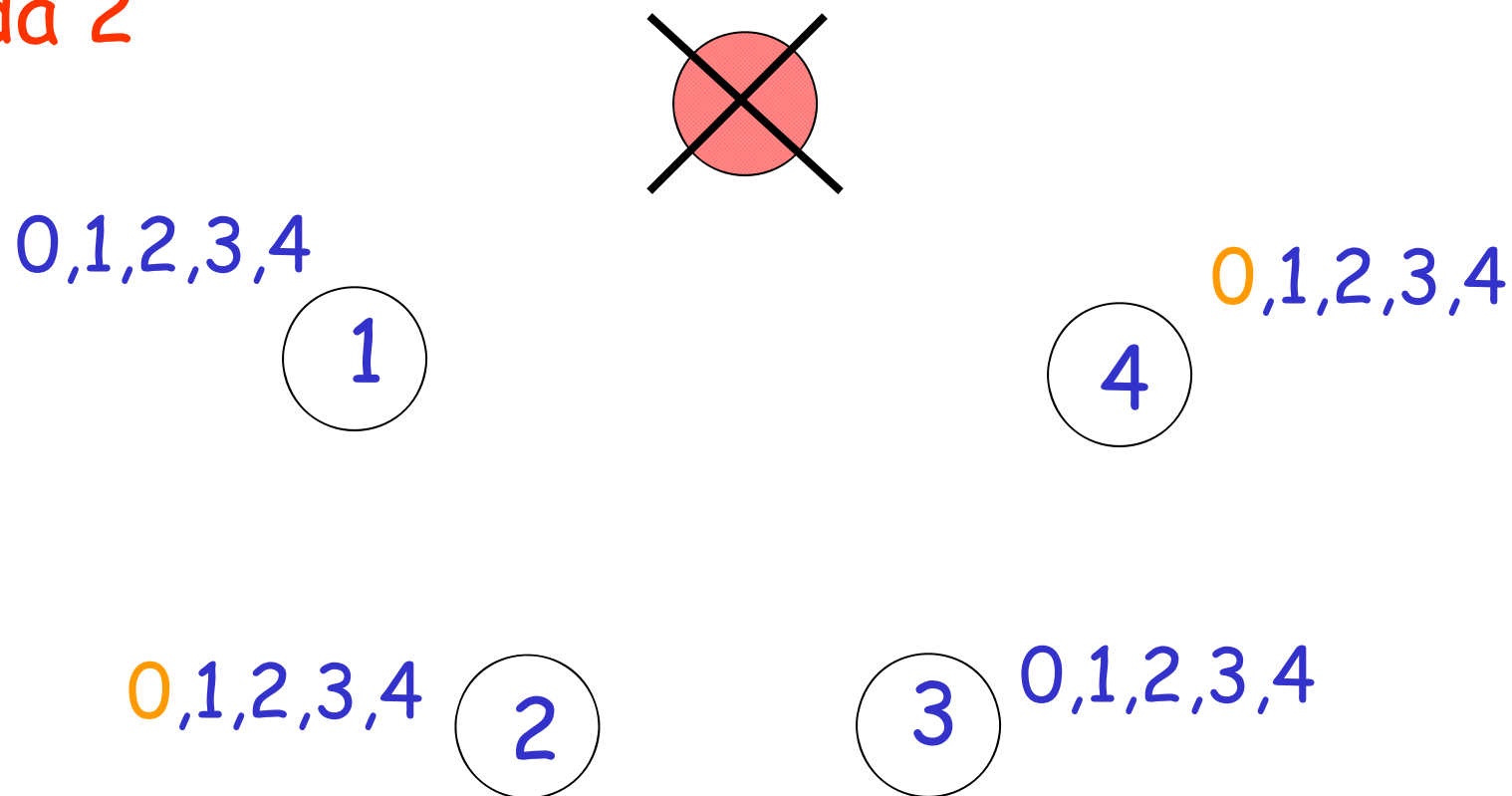
Runda 1



Pošalji svima sve vrednosti

Primer: $f=1$ otkaza, $f+1 = 2$ potrebne runde

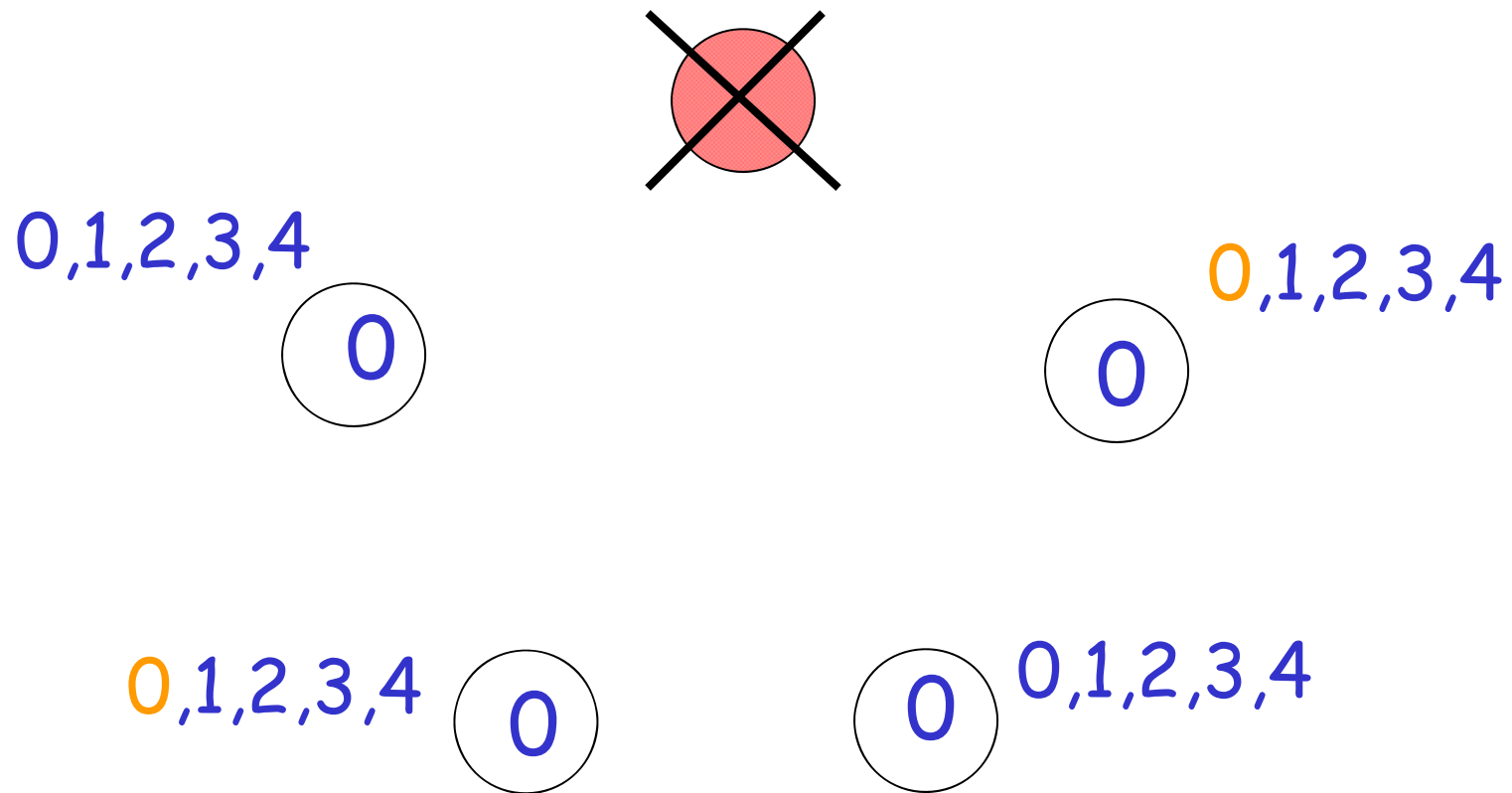
Runda 2



Pošalji svima sve vrednosti

Primer: $f=1$ otkaza, $f+1 = 2$ potrebne runde

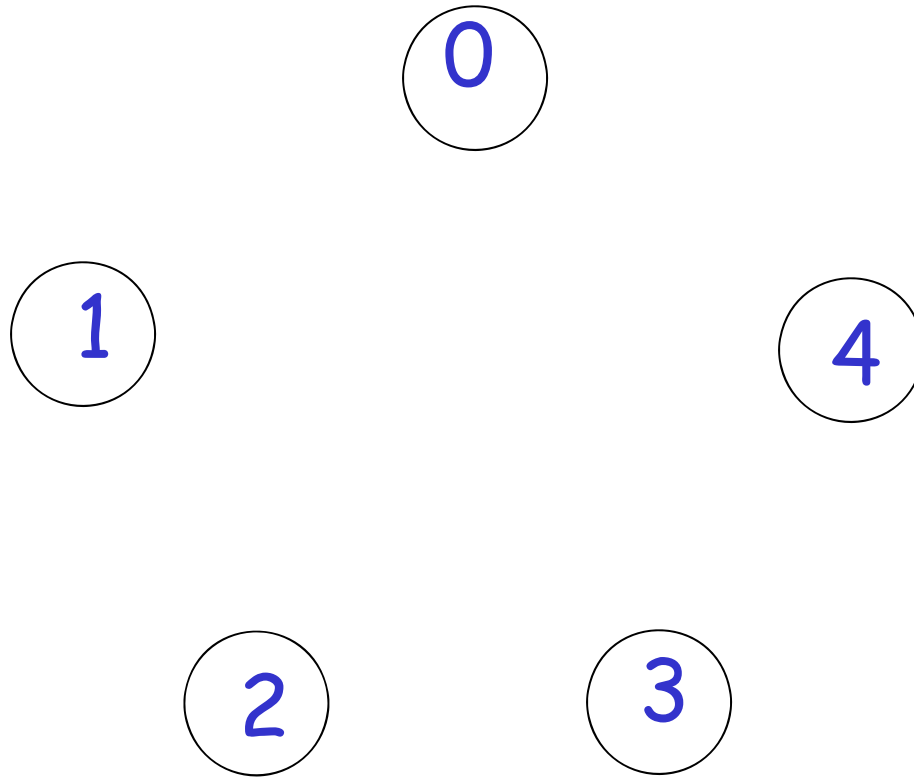
Kraj



Odluči se za min vrednost

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

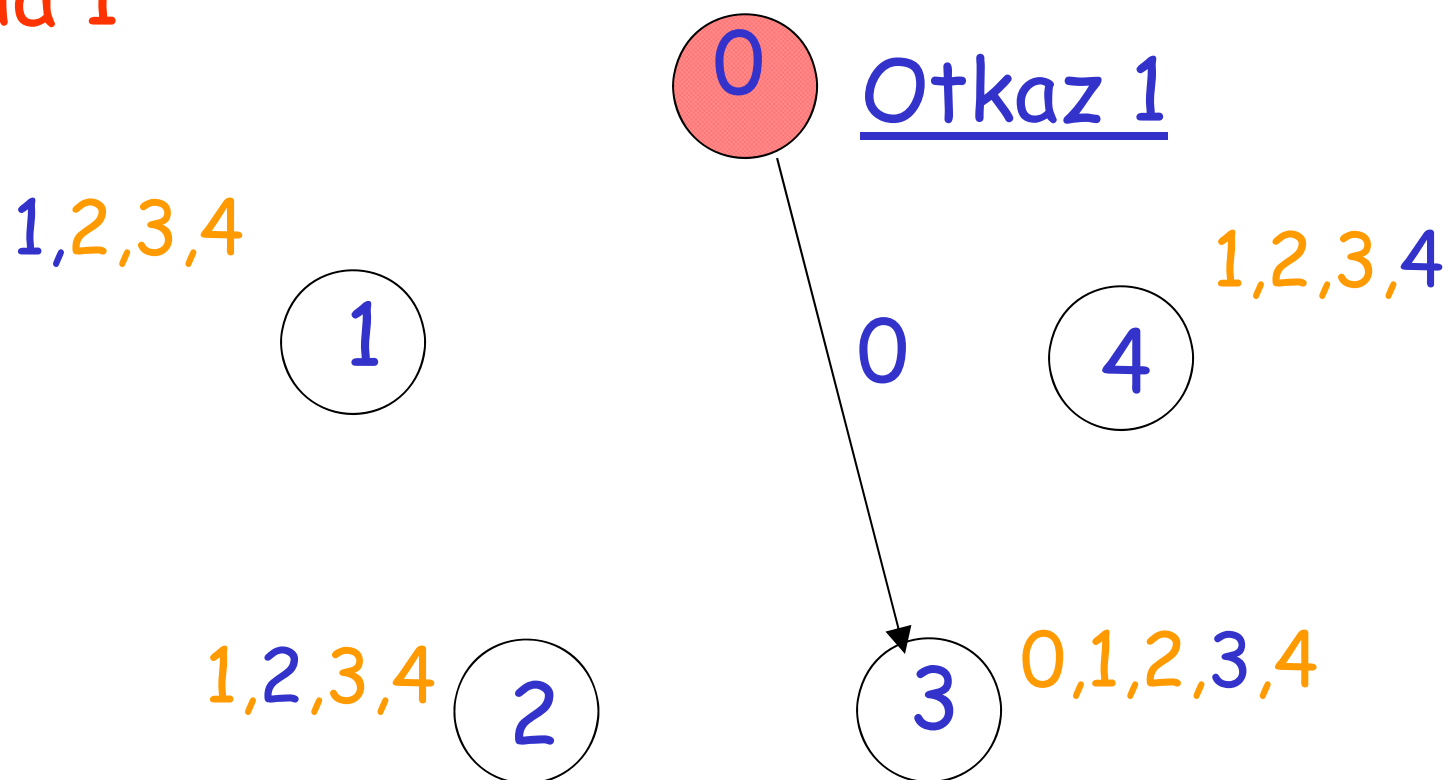
Početak



Drugi primer: izvršenje sa 2 otkaza

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

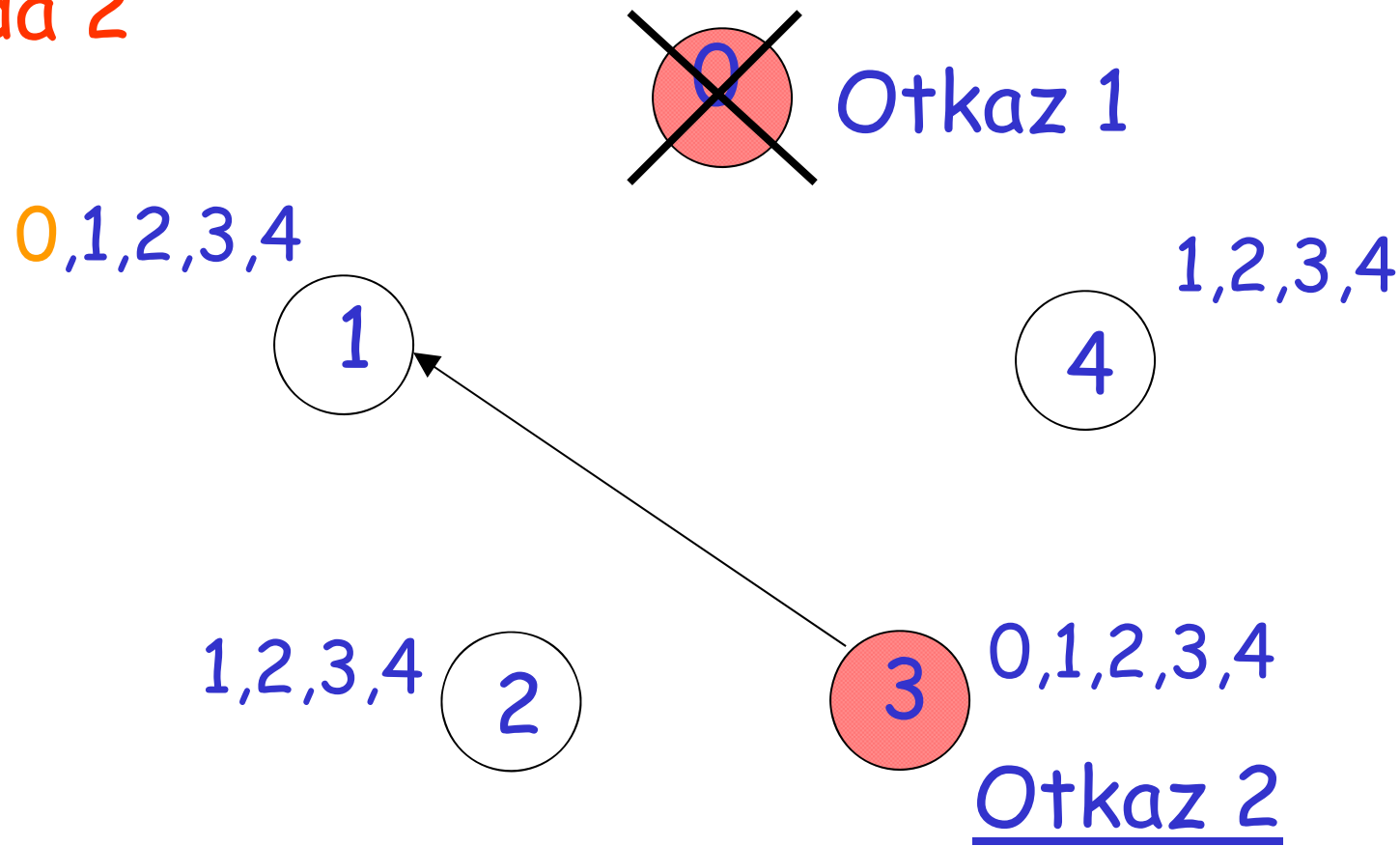
Runda 1



Pošalji svima sve vrednosti

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

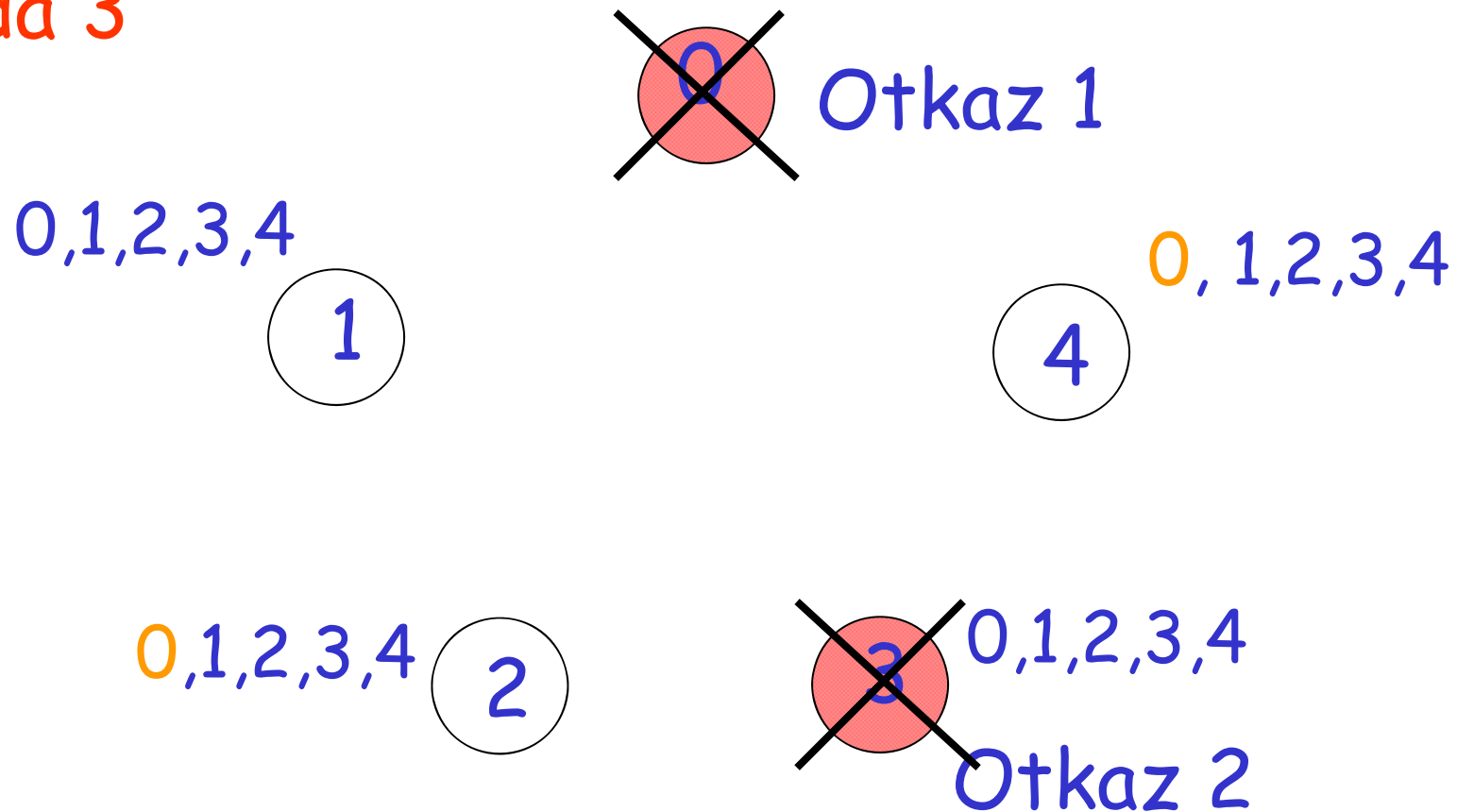
Runda 2



Pošalji svima sve vrednosti

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

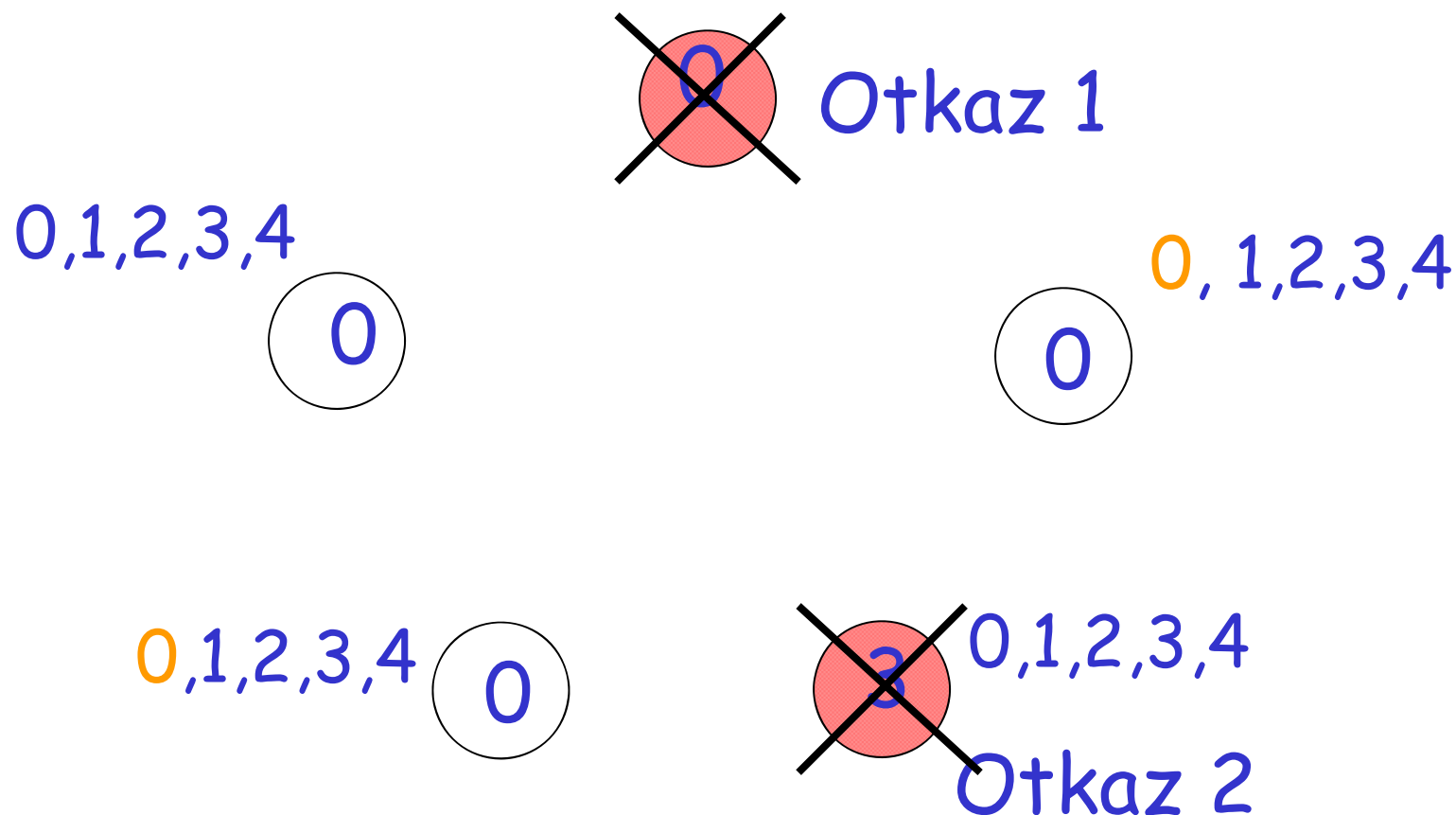
Runda 3



Pošalji svima sve vrednosti

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

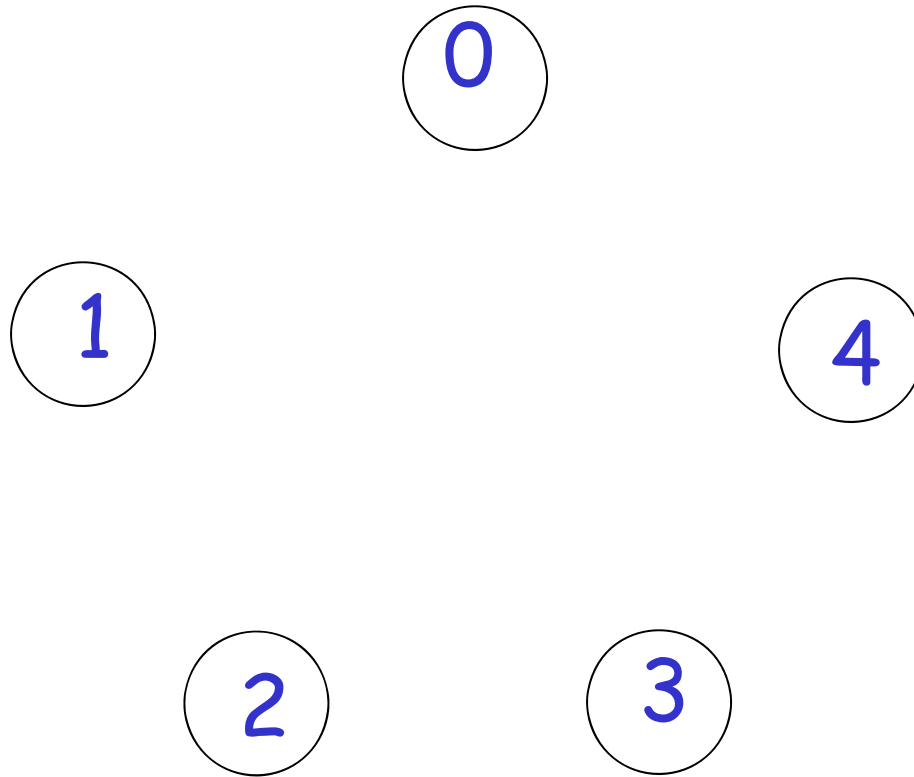
Kraj



Odluči se za min vrednost

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

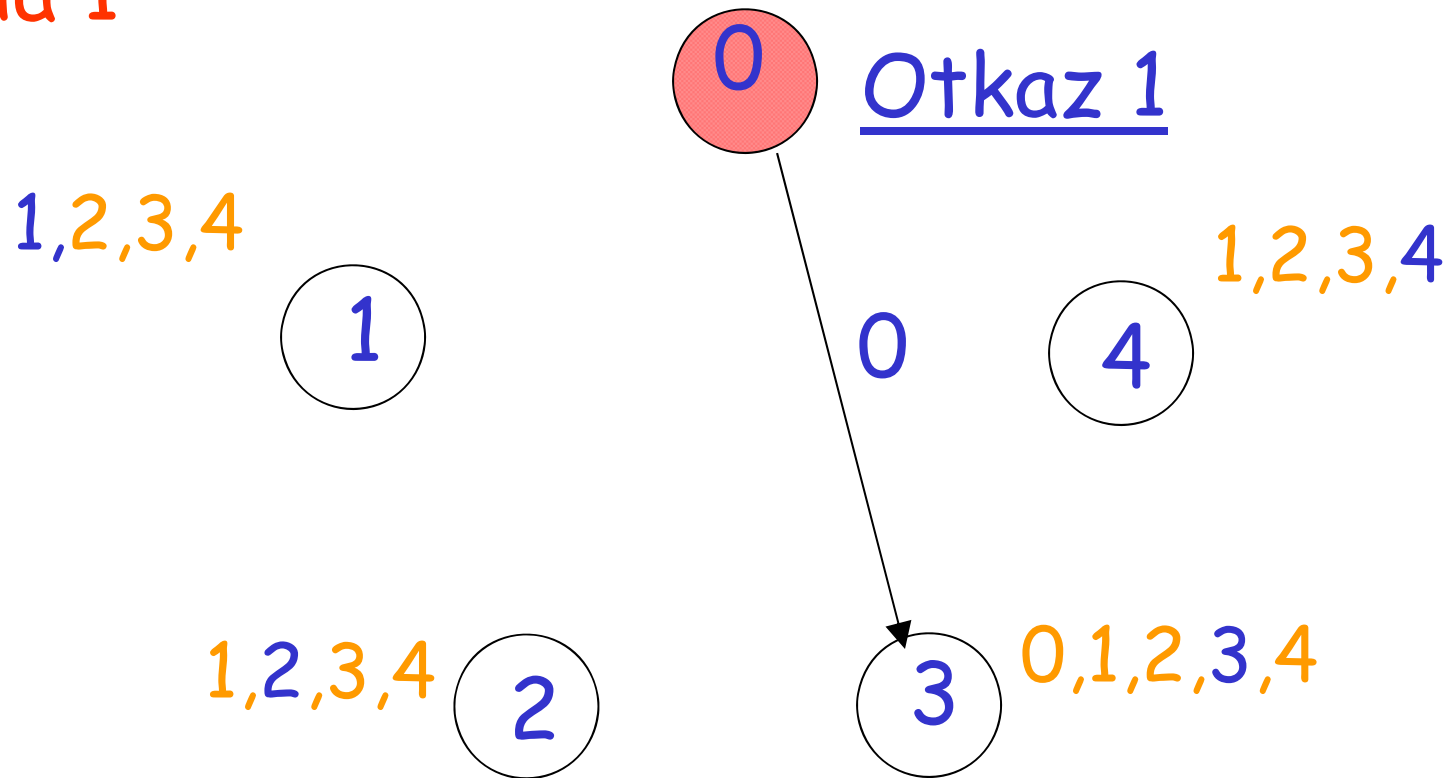
Početak



Još jedan primer izvršenja sa 2 otkaza

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

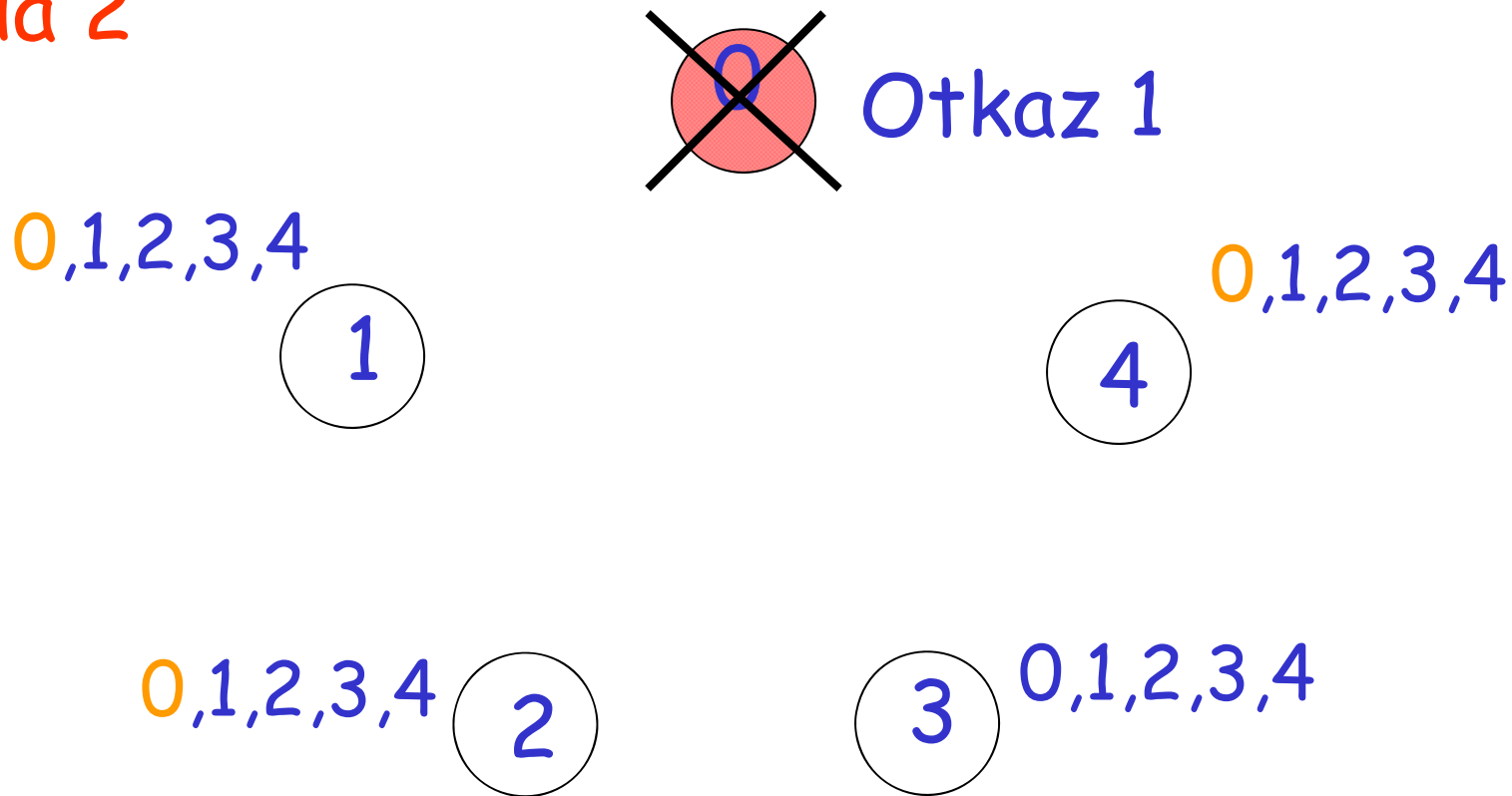
Runda 1



Pošalji svima sve vrednosti

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

Runda 2

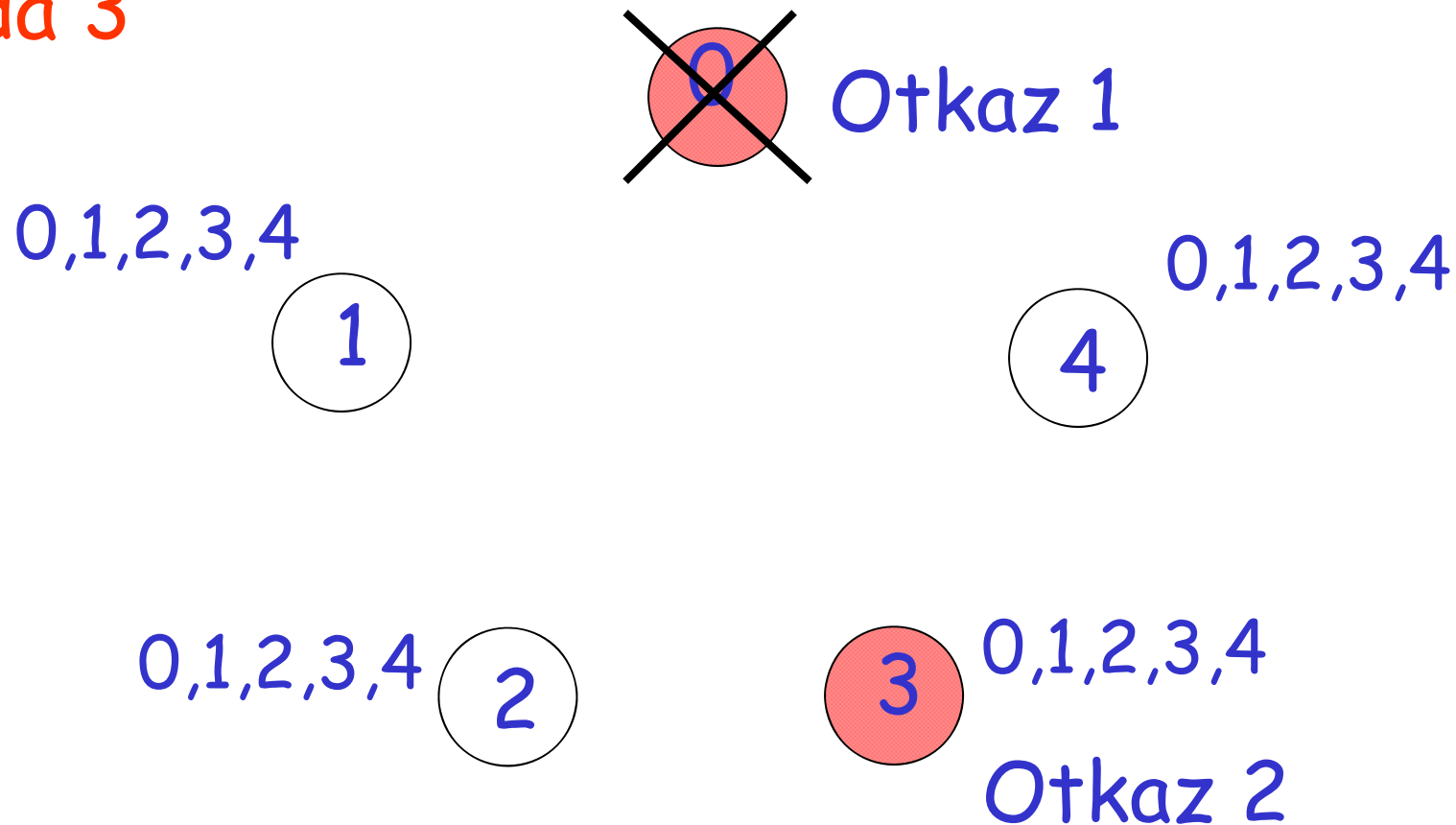


Pošalji svima sve vrednosti

Napomena: Na kraju ove runde svi procesi
znaju sve druge vrednosti

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

Runda 3

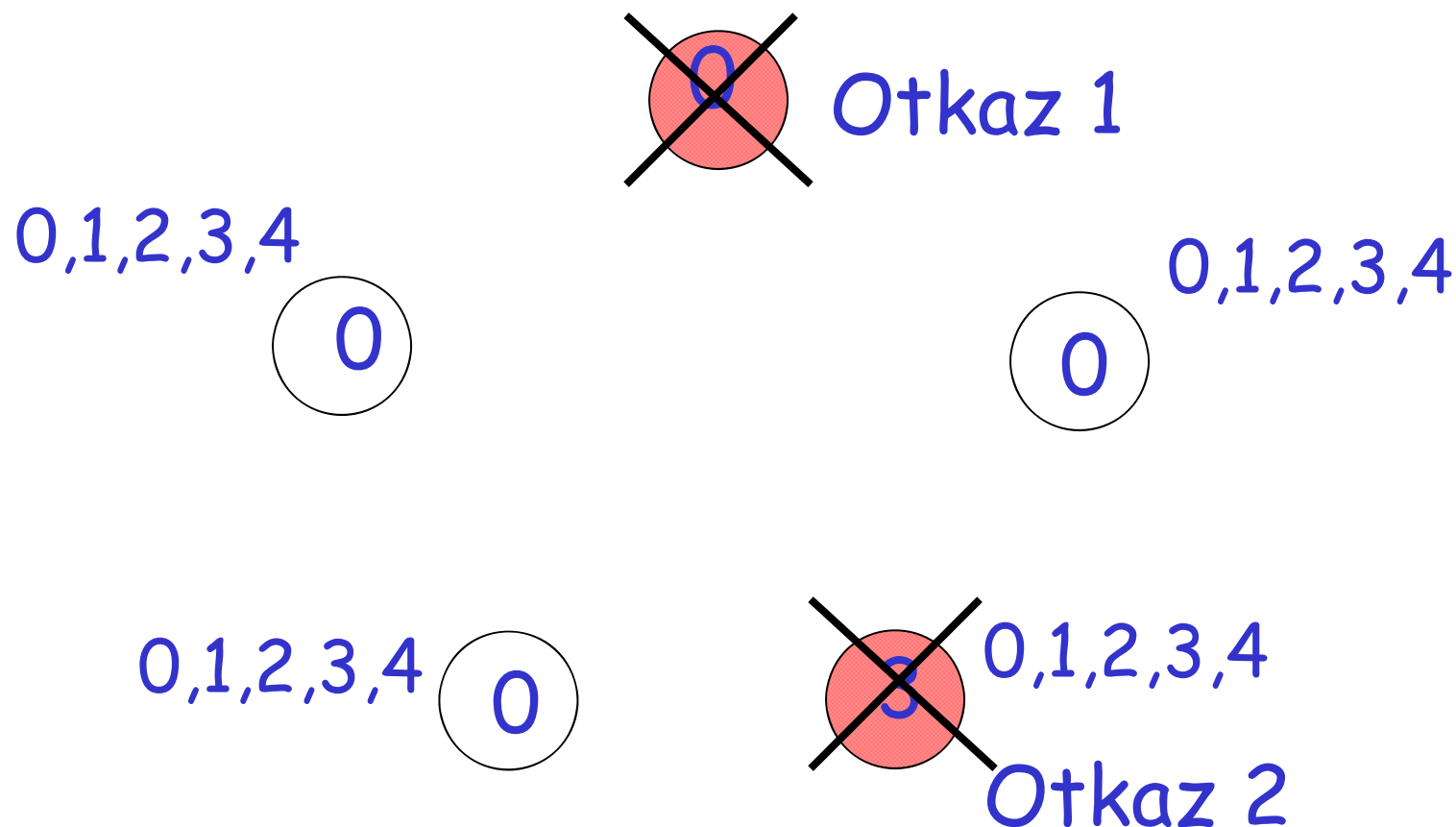


Pošalji svima sve vrednosti

(niko ne dobija novu vrednost u ovoj rundi)

Primer: $f=2$ otkaza, $f+1 = 3$ potrebne runde

Kraj

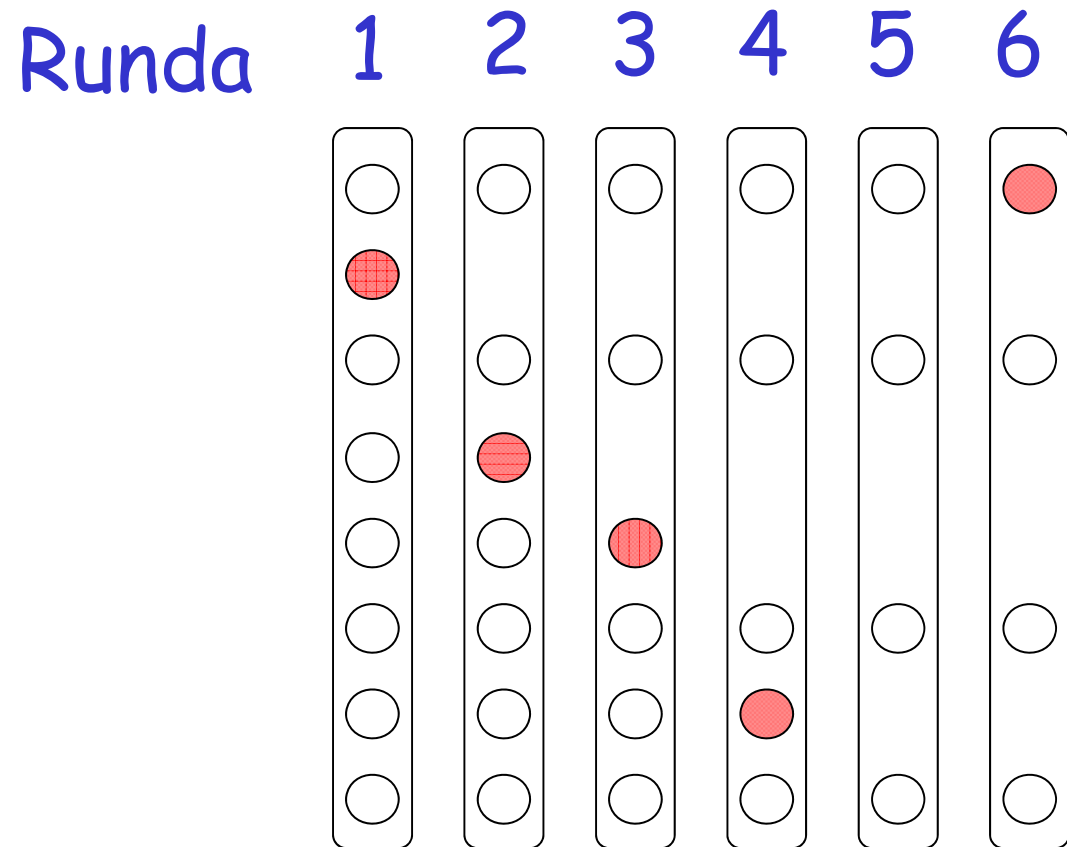


Odluči se za min vrednost

Ako ima f otkaza i $f+1$ rundi, onda
postoji runda bez otkaza procesa

Primer:

5 otkaza,
6 rundi



Bez otkaza

U algoritmu, na kraju
runde bez otkaza:

- Svaki (ispravan) proces zna sve vrednosti svih drugih procesa koji učestvuju
- To znanje se ne menja sve do kraja algoritma

Zbog toga, na kraju
runde bez otkaza:

svi bi se odlučili za istu vrednost

Ali, pošto nije poznata tačna pozicija
ove runde, moramo pustiti algoritam
da izvrši svih $f+1$ rundi

Validnost algoritma:

kad svi procesi počnu sa jedom istom
ulaznom vred. onda je konsenzus ta vred.

Ovo važi, pošto je vrednost za koju se odluči
svaki proces neka ulazna vrednost

Donja granica

Teorema: Bilo koji f -elastičan alg. konsenzusa
zahteva bar $f+1$ rundi

Skica dokaza:

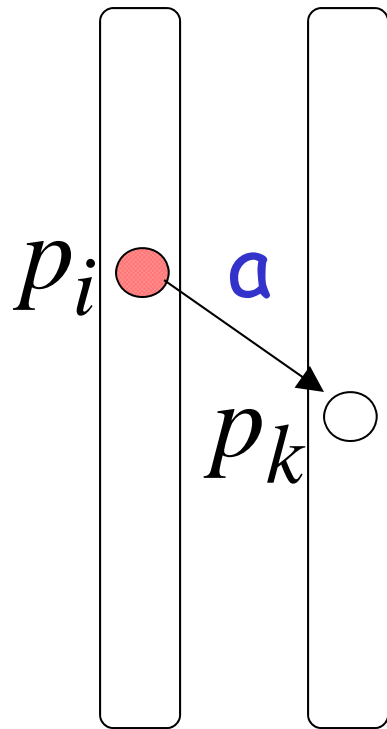
Predpost. zbog kontradikcije da je f
ili manje rundi dovoljno

Scenario najgoreg slučaja:

U svakoj rundi,
postoji proces koji otkazuje

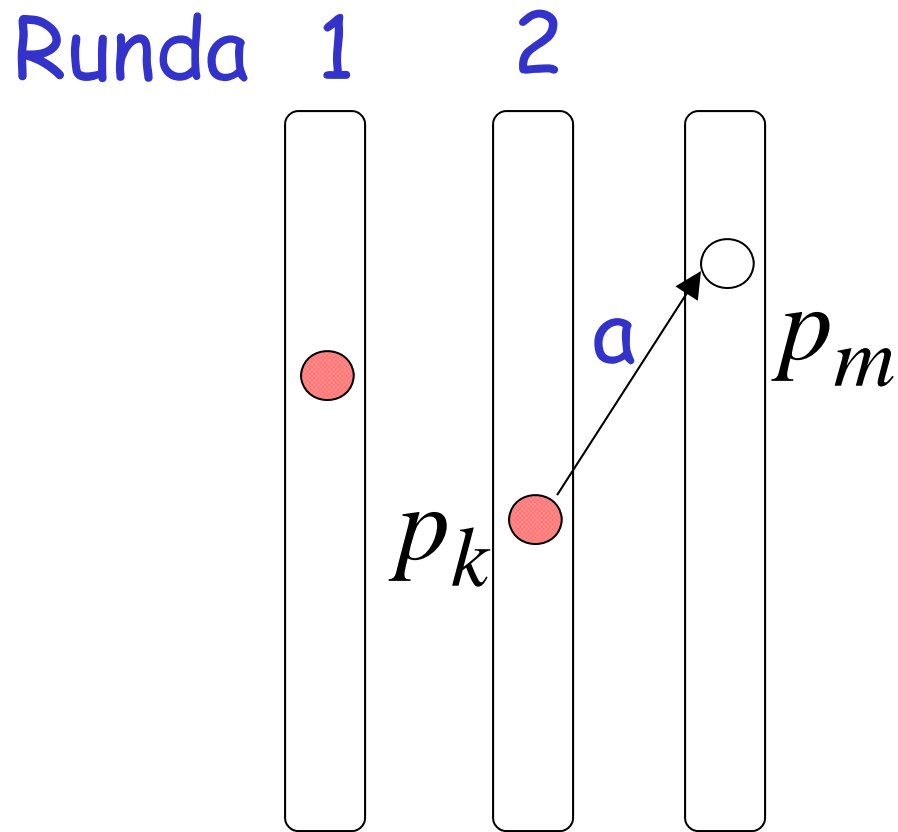
Scenario najgoreg slučaja

Runda 1



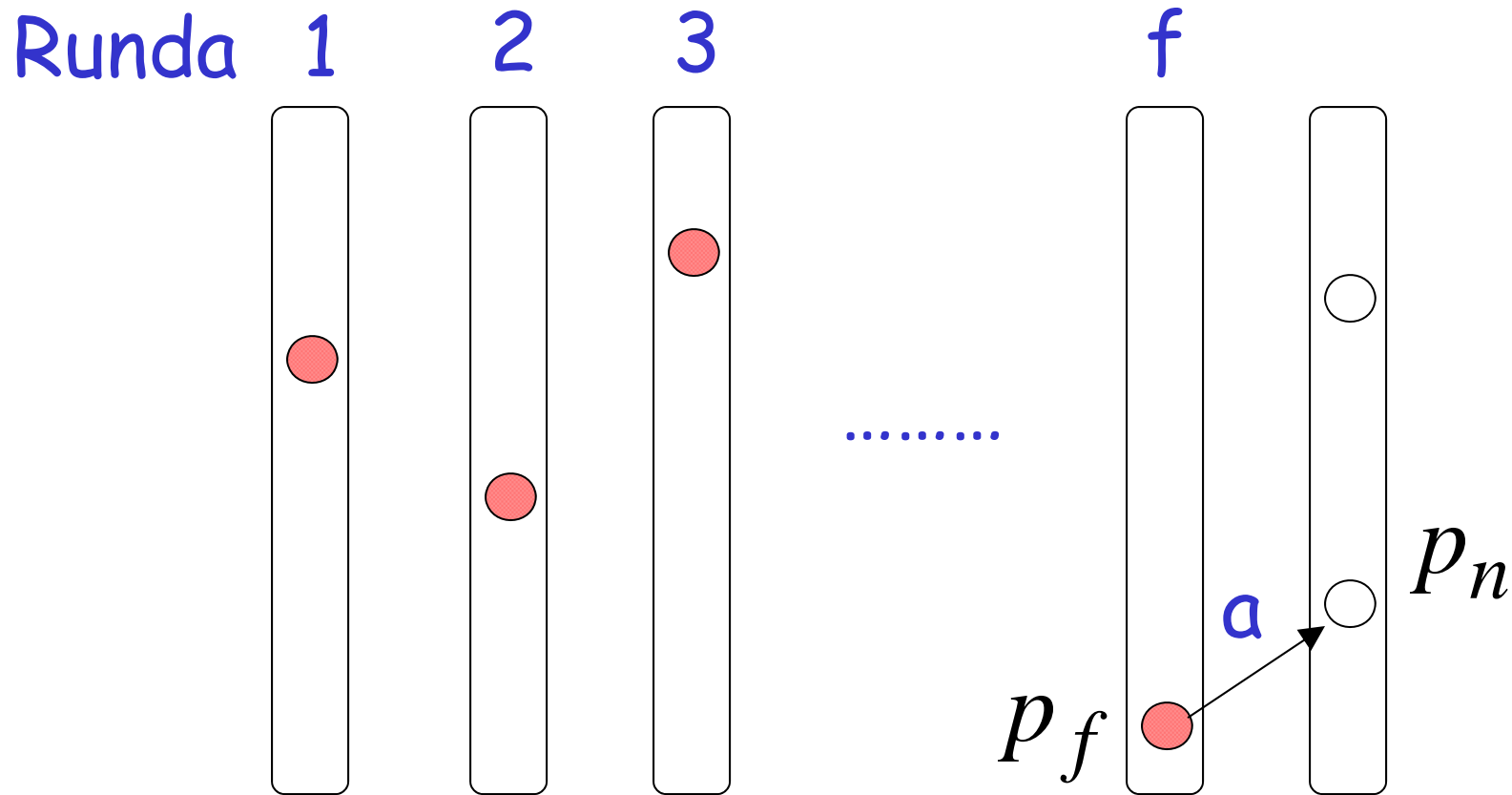
pre nego proces p_i otkaže, on šalje svoju vred. a samo procesu p_k

Scenario najgoreg slučaja



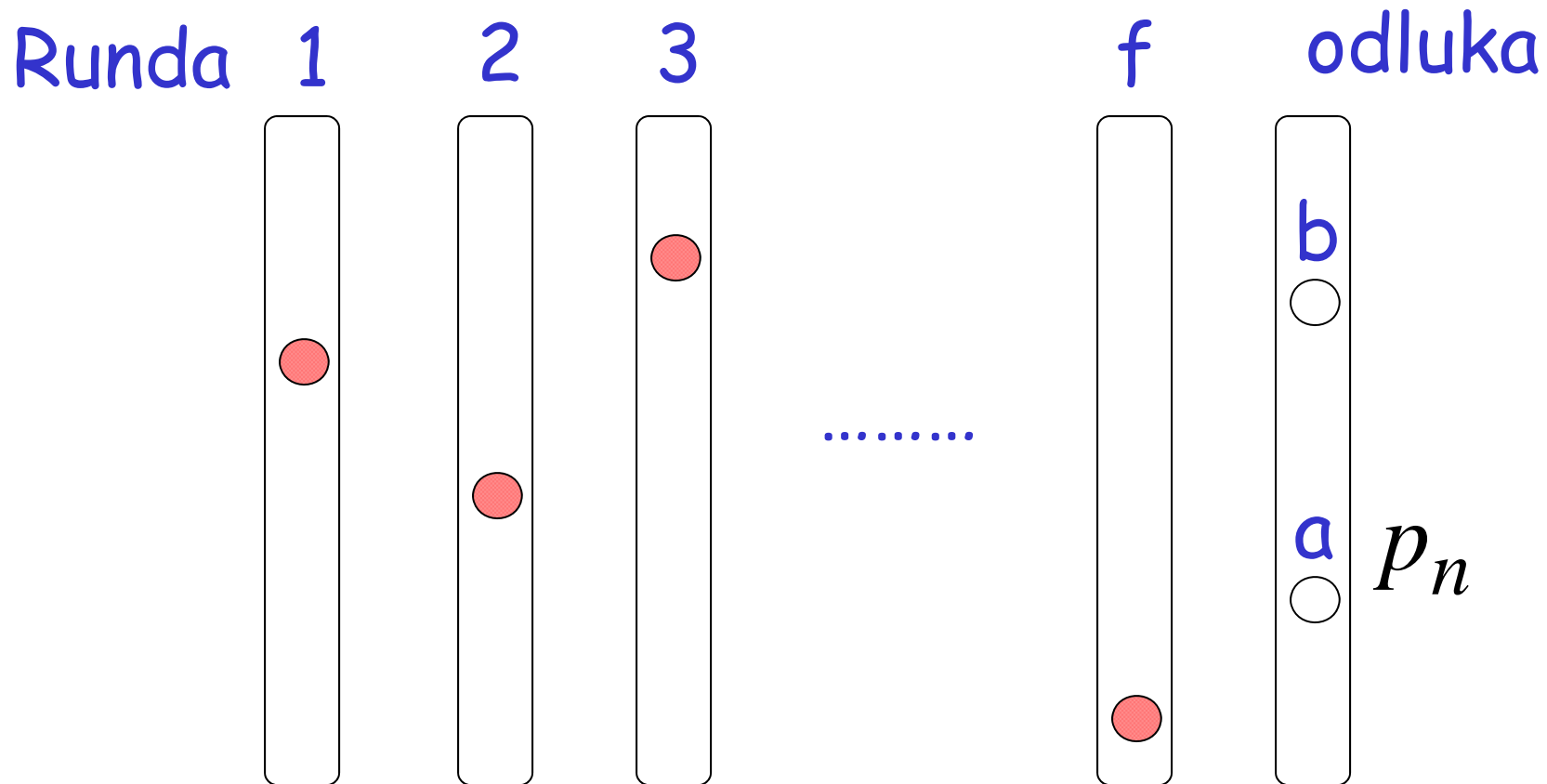
pre nego proces p_k otkaže, on šalje vred. a
samo procesu p_m

Scenario najgoreg slučaja



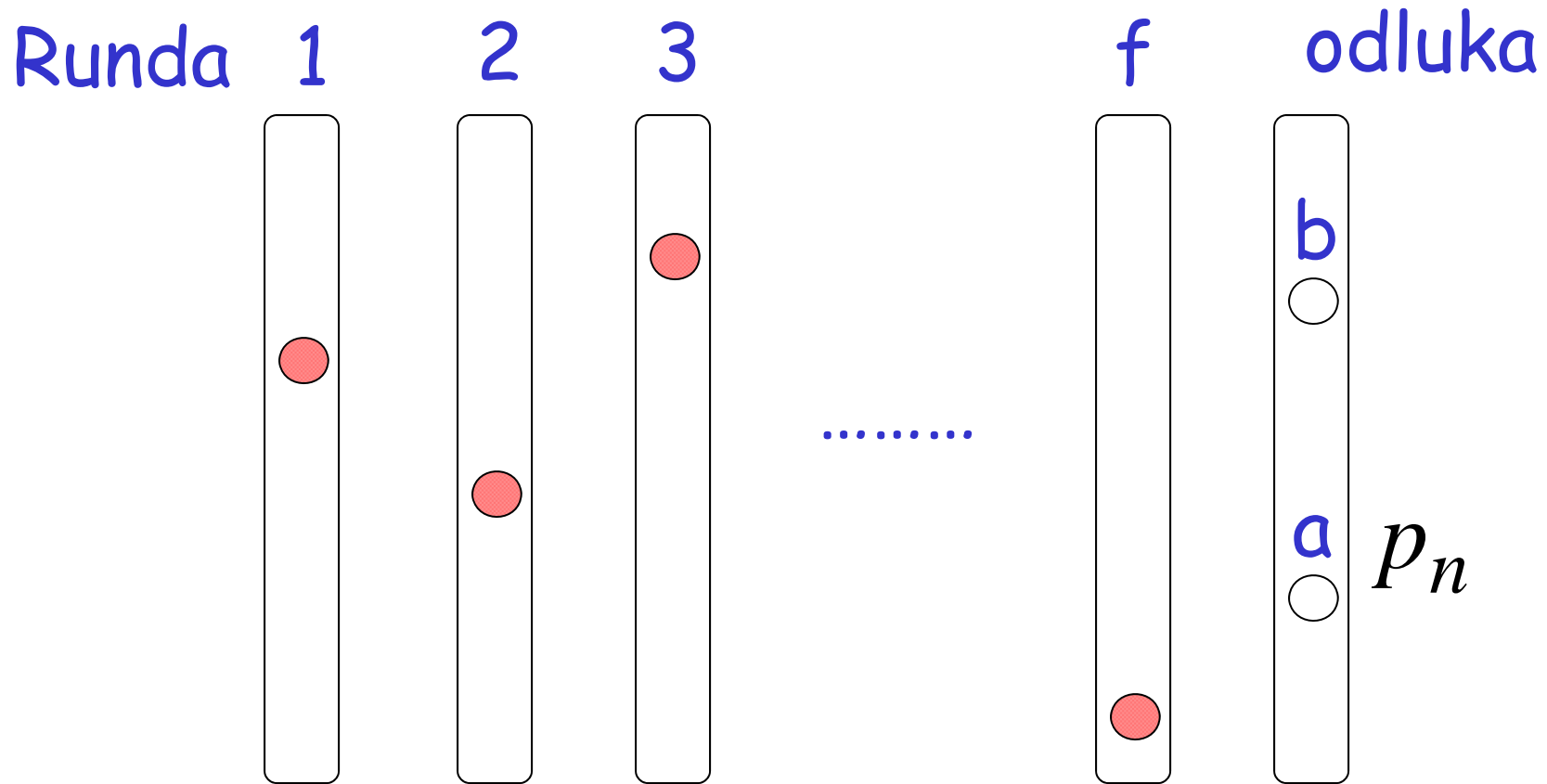
Na kraju runde f samo proces p_n
zna za vrednost a

Scenario najgoreg slučaja



proces p_n može odlučiti a, a svi drugi procesi mogu odlučiti drugu vrednost (b)

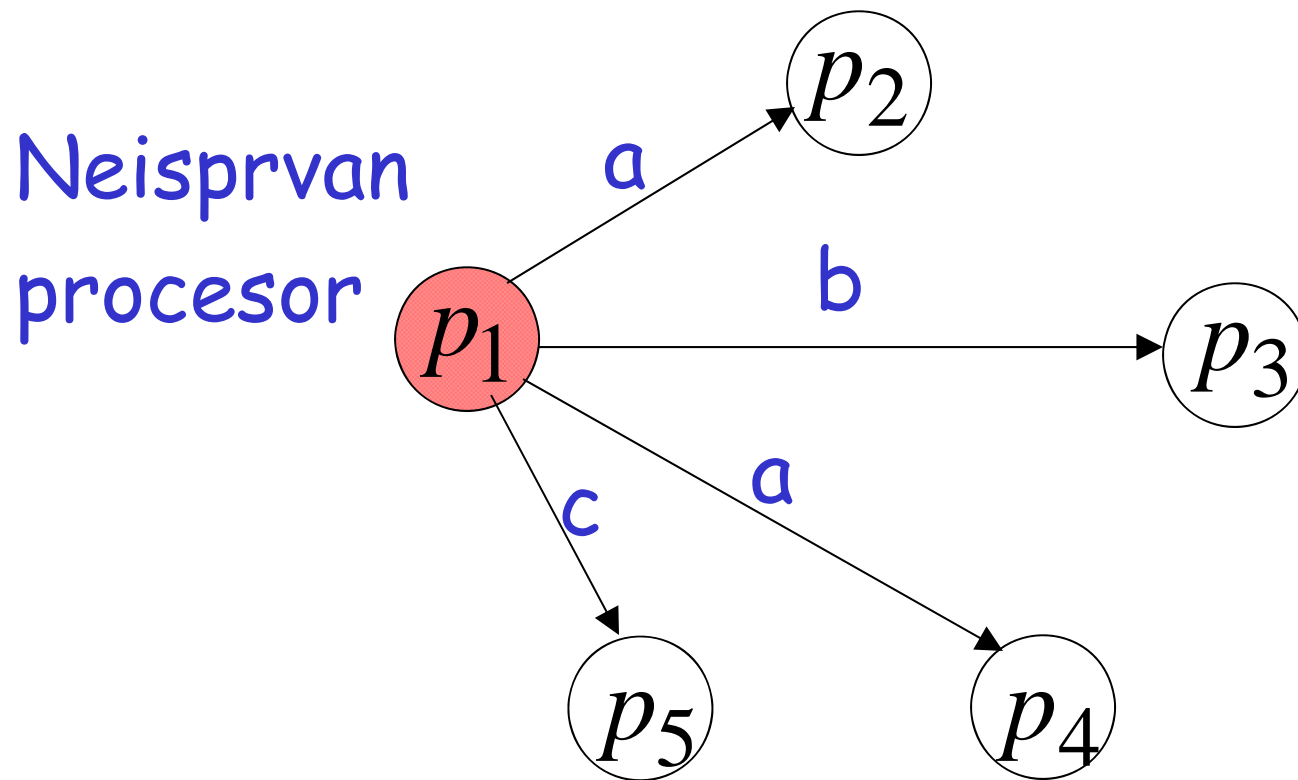
Scenario najgoreg slučaja



Zbog toga f rundi nije dovoljno
Potrebno je bar $f+1$ rundi

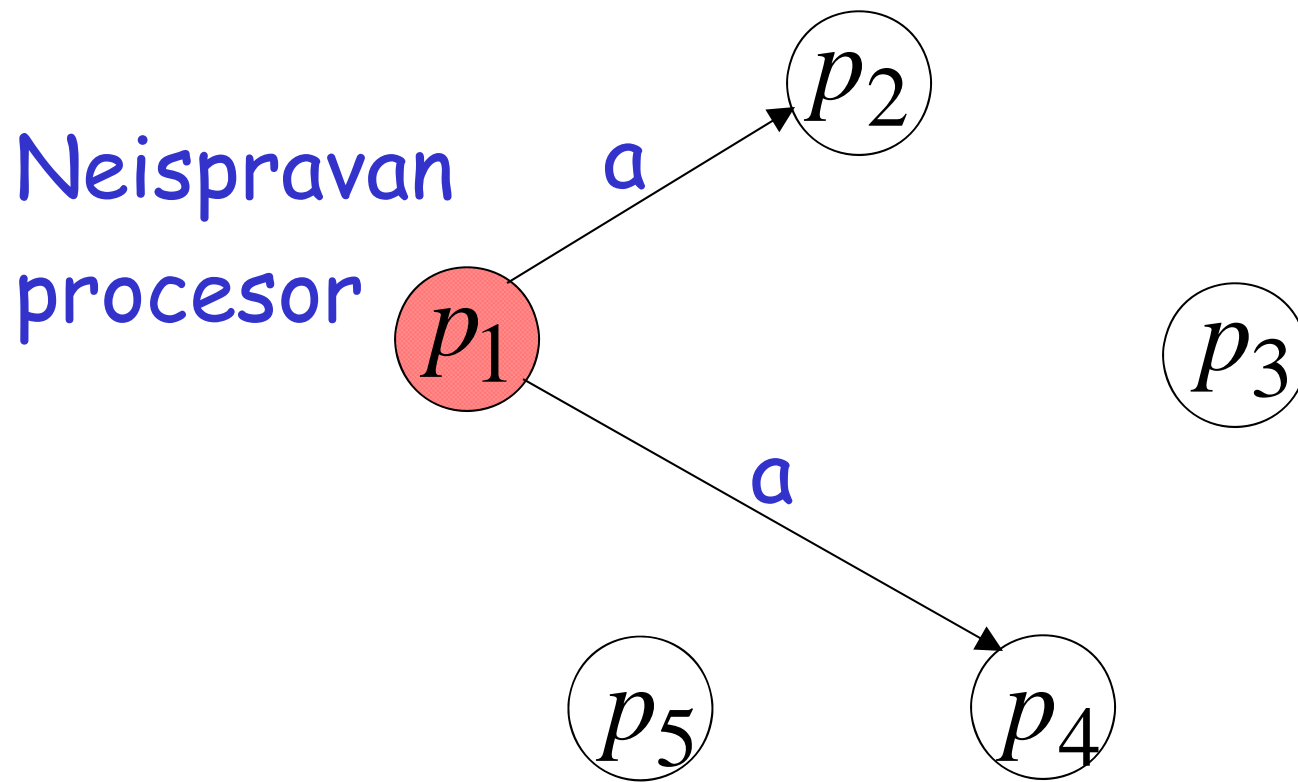
Vizantijski otkazi (Byzantine Failures)

Vizantijski otkazi

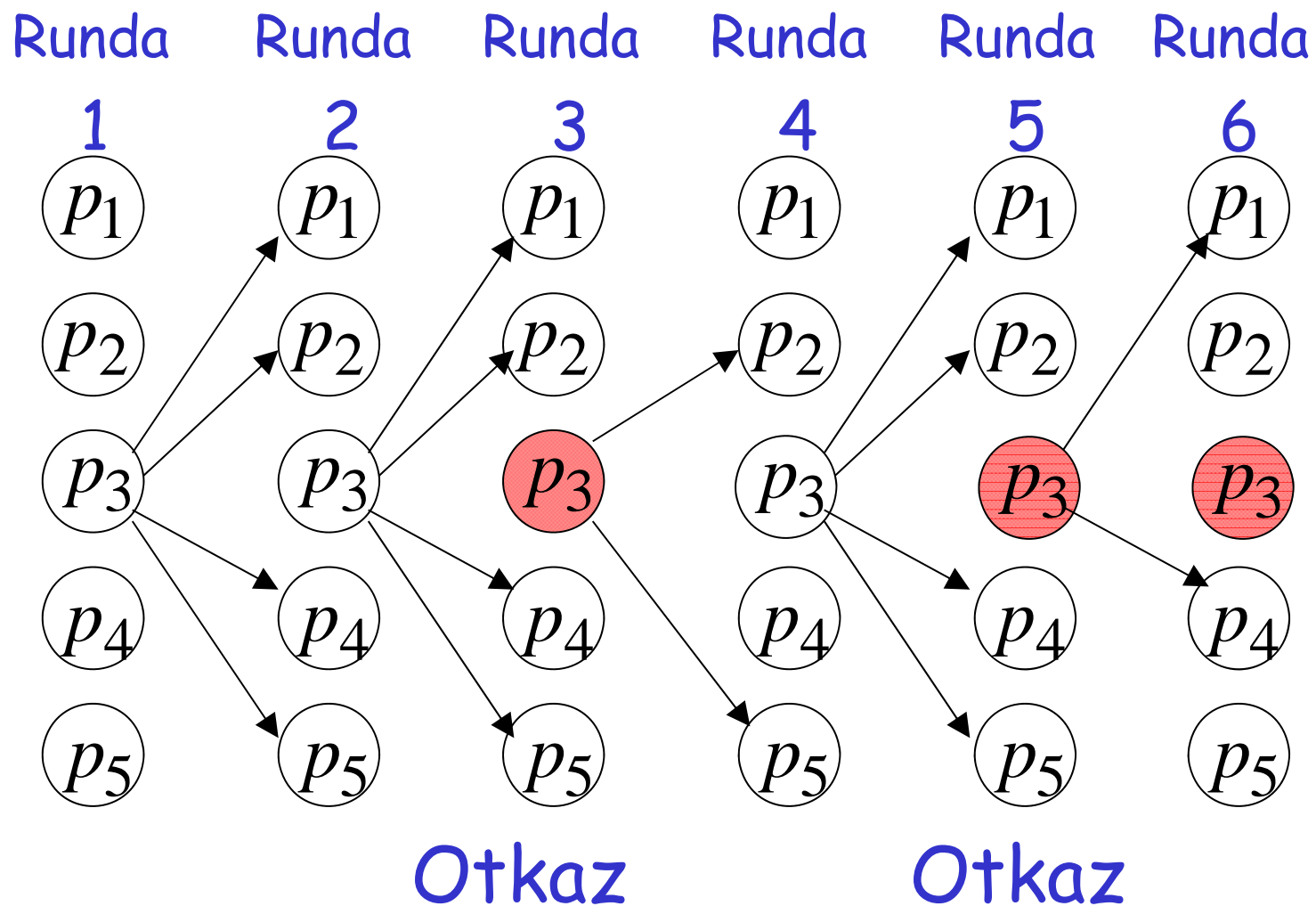


Različiti procesi primaju različite vrednosti

Neke poruke mogu biti izgubljene



Vizantijski proces se može ponašati kao proces koji je ispao iz rada (Crash)



Nakon otkaza, proces nastavlja da funkcioniše u mreži

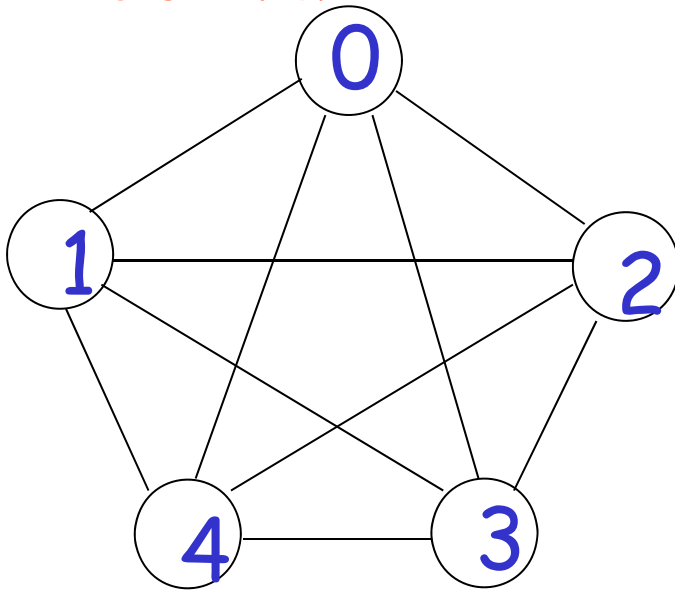
Konsenzus sa vizantijskim otkazima

f-elastičan algoritam konsenzusa:

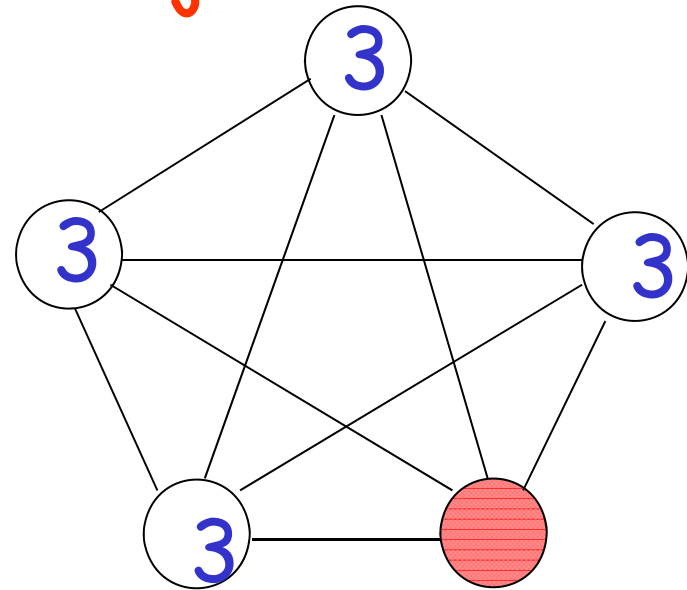
rešava konsenzus za f procesa u otkazu

Primer: Ulaz i izlaz za
1-elastičan algoritam konsenzusa

Početak



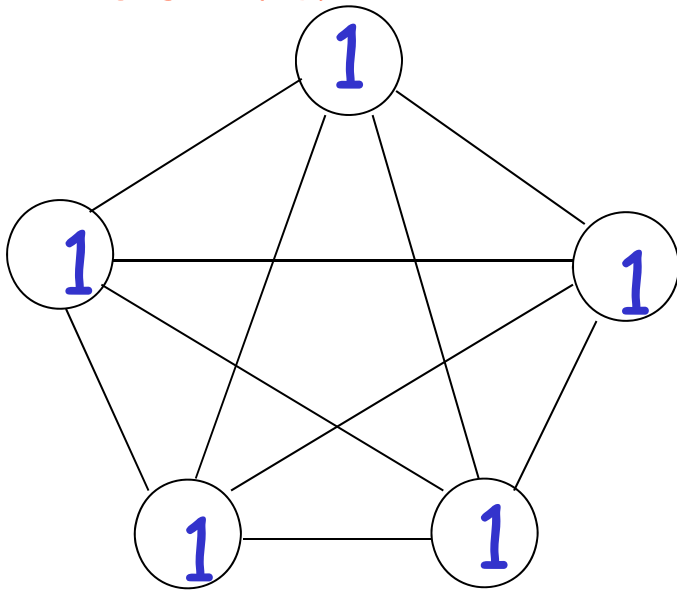
Kraj



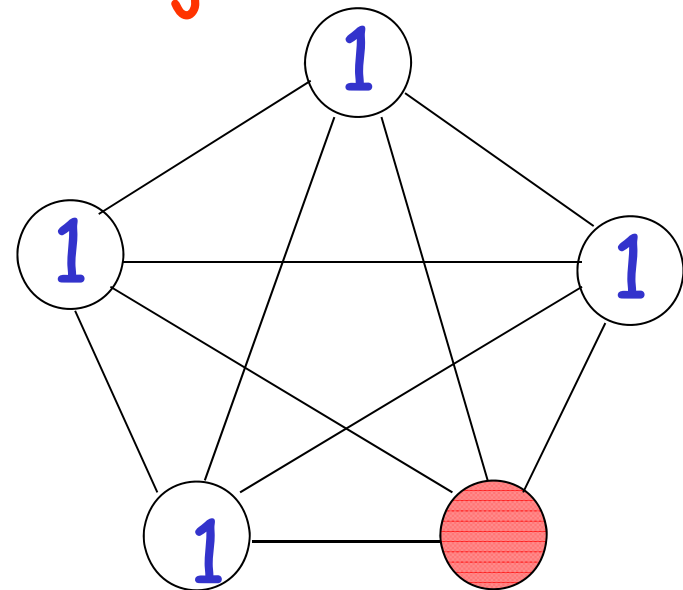
Uslov validnosti:

ako svi ispravni procesi počnu sa istom vred. onda se svi ispravni procesi odlučuju za tu vrednost

Početak



Kraj



Donja granica za broj rundi

Teorema: Bilo koji f -elastičan alg. konsenzusa sa vizantijskim otkazima zahteva bar $f+1$ rundi

Dokaz:

sledi iz donje granice za ispade (crash)

Jedan algoritam konsenzusa

Algoritam Kralj (King)

rešava konsenzus za

n procesa i

f otkaza, gde je $f < \frac{n}{4}$

Algoritam Kralj

Postoji $f + 1$ faza

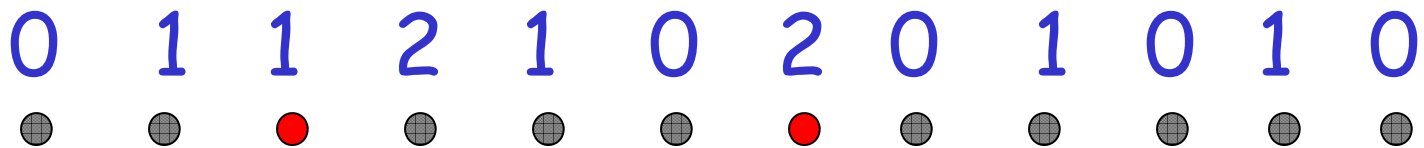
Svaka faza ima dve runde slanja svima

U svakoj fazi postoji različit kralj

Primer: 12 procesa, 2 otkaza, 3 kralja

početne vrednosti

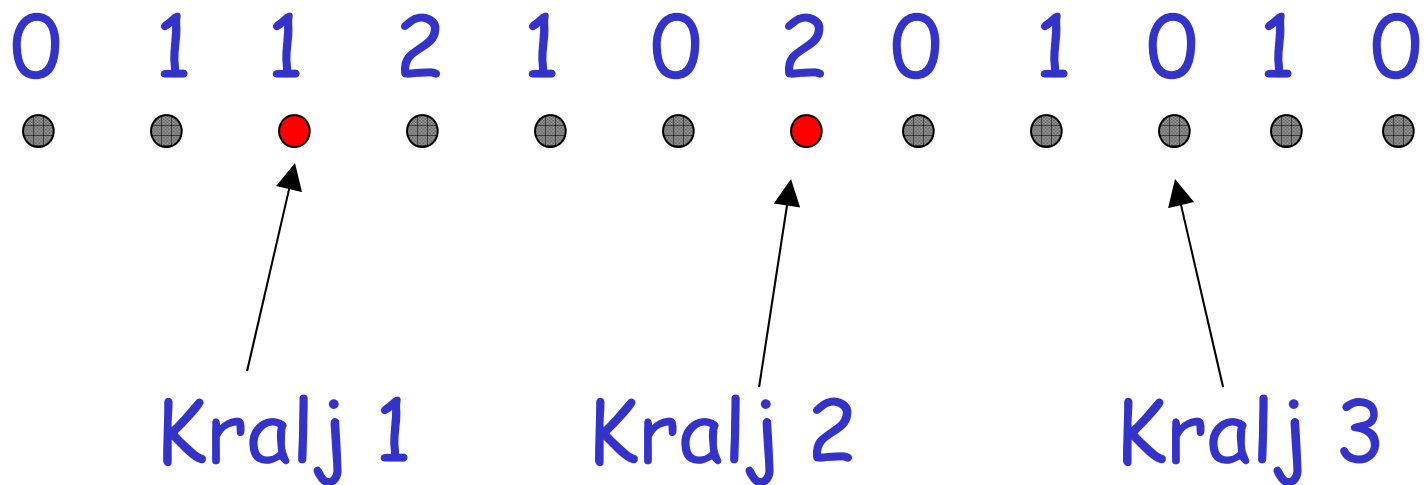
0 1 1 2 1 0 2 0 1 0 1 0



Otkazali

Primer: 12 procesa, 2 otkaza, 3 kralja

početne vrednosti



Napomena: Postoji kralj koji je ispravan

Algoritam Kralj

Svaki procesor P_i ima prioritetnu vrednost v_i

Na početku, prioritetna vrednost se postavlja na početnu vrednost

Algoritam Kralj

Faza k

Runda 1, procesor p_i :

- Šalji svima prioritetnu vred. v_i
- Neka je a većinska vrednost od primljenih vred. (uključujući v_i)
(u slučaju nerešenog ishoda izaberi proizvoljnu vred.)
- Postavi $v_i = a$

Algoritam Kralj

Faza k

Runda 2, kralj p_k :

Šalji svima novu prior. vrednost v_k

Runda 2, proces p_i :

Ako v_i ima većinu, ne veću od $\frac{n}{2} + f$

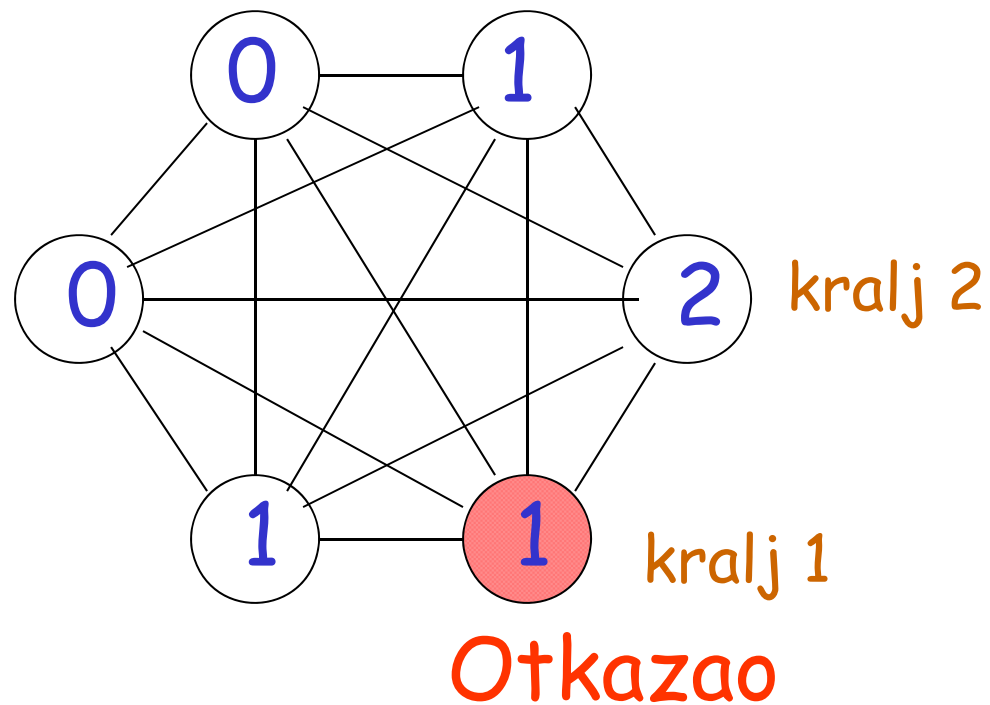
onda postavi $v_i = v_k$

Algoritam Kralj

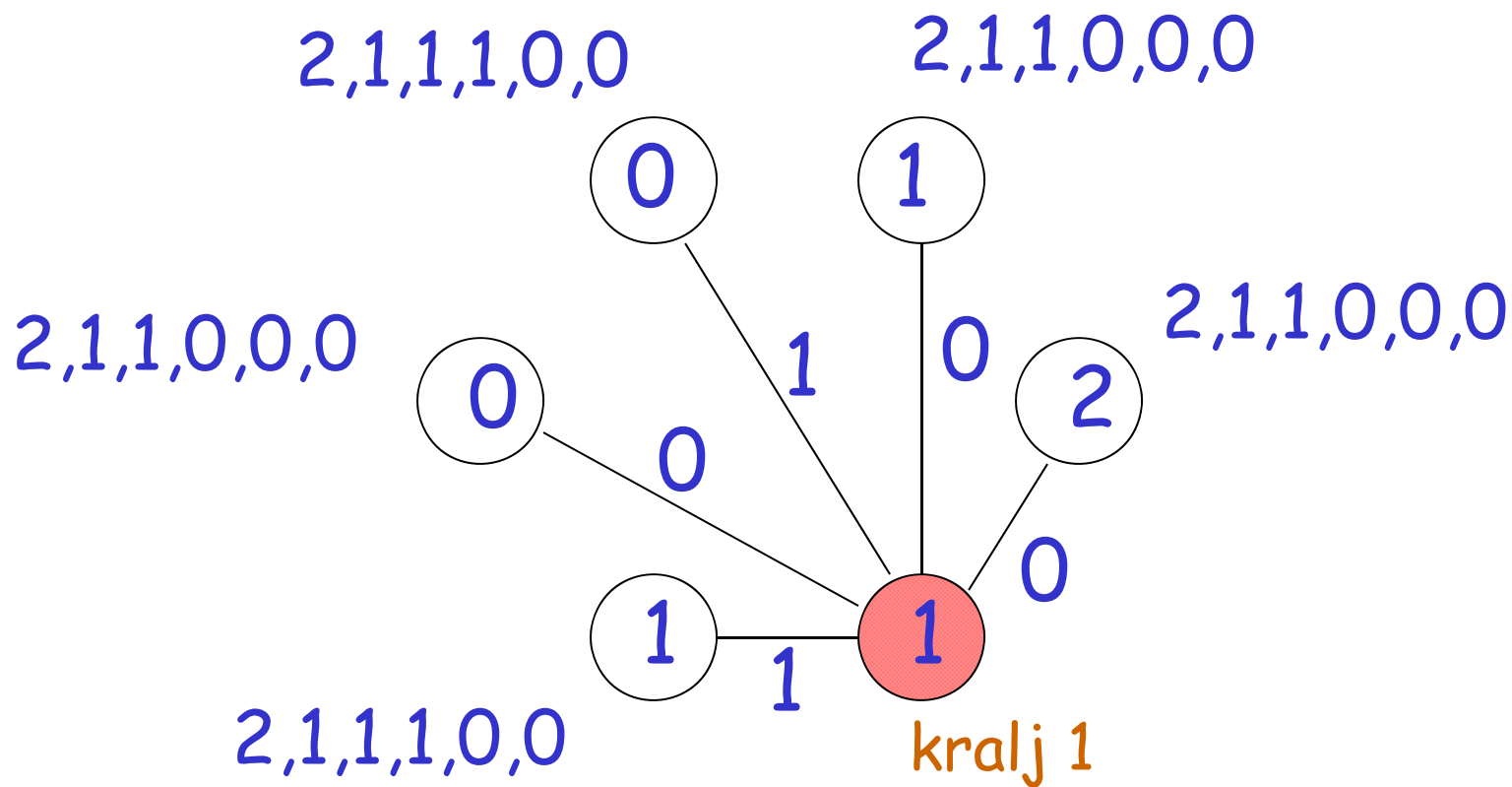
Kraj Faze $f+1$:

Svaki proces odlučuje o prior. vrednosti

Primer: 6 procesa, 1 otkaz



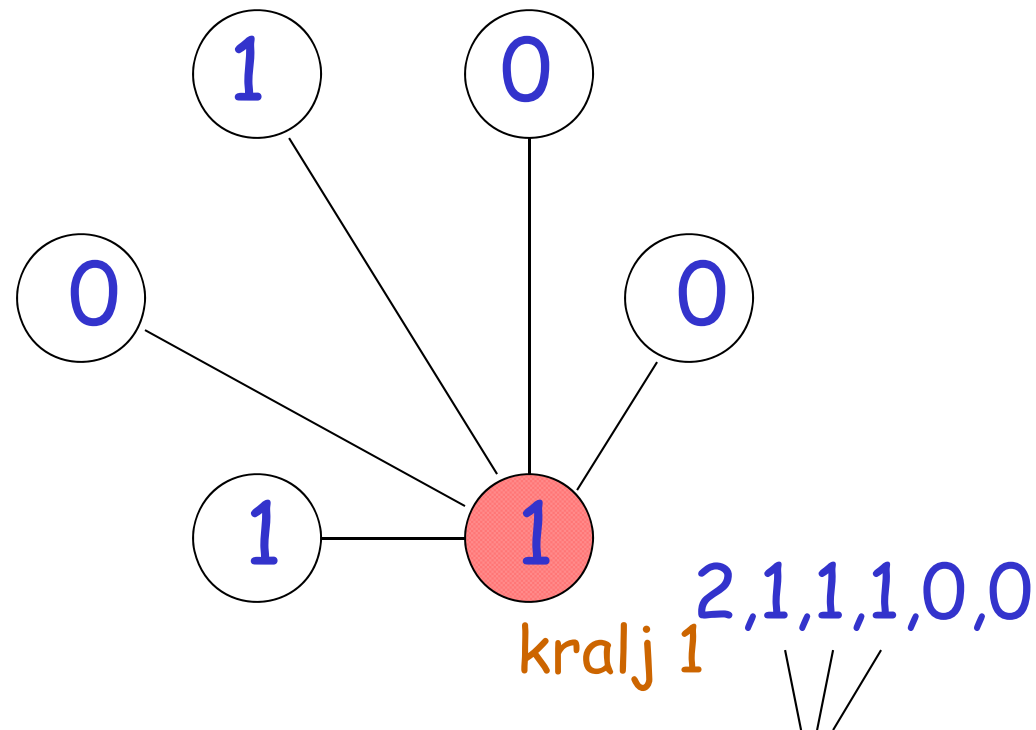
Faza 1, Runda 1



Svi šalju svima

Faza 1, Runda 1

Izaberi većinsku vred

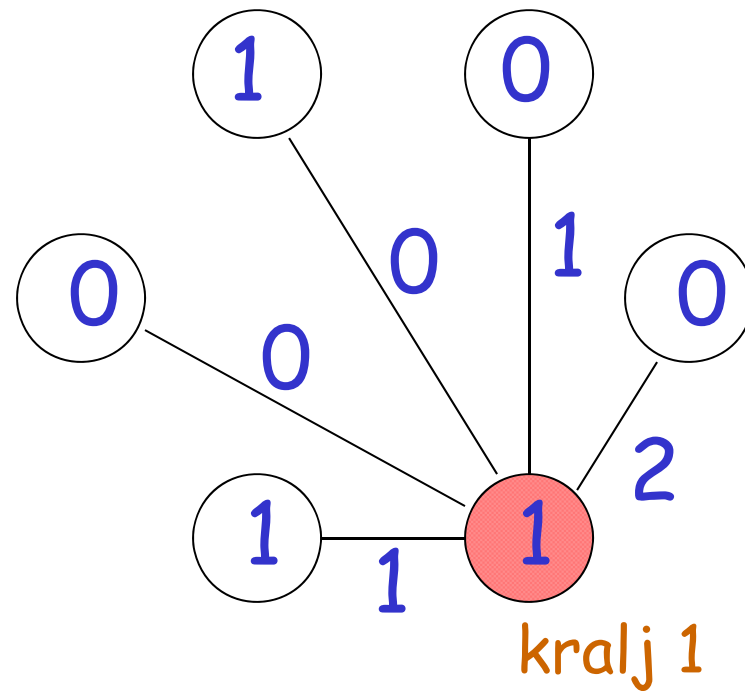


Svima većinski glas bio

$$3 < \frac{n}{2} + f = 4$$

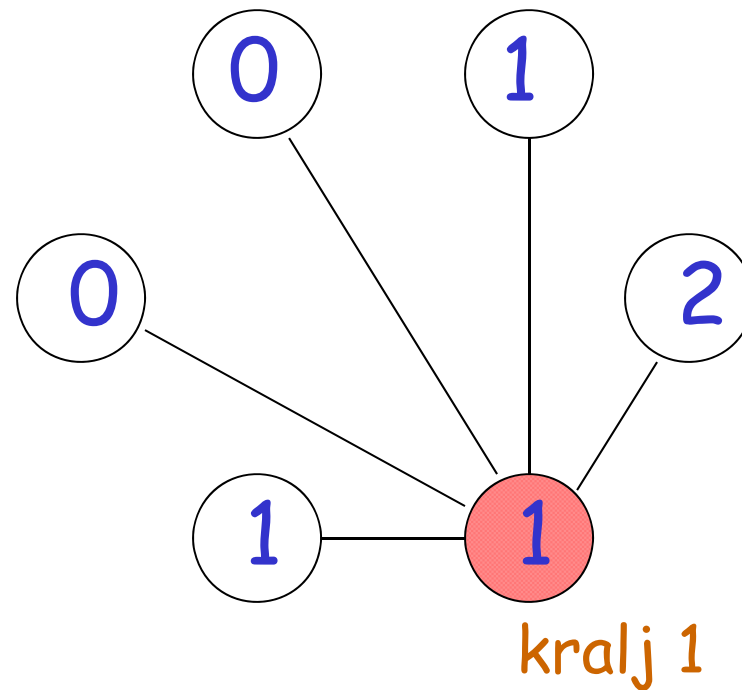
U rundi 2, svi će izabrati kraljevu vrednost

Faza 1, Runda 2



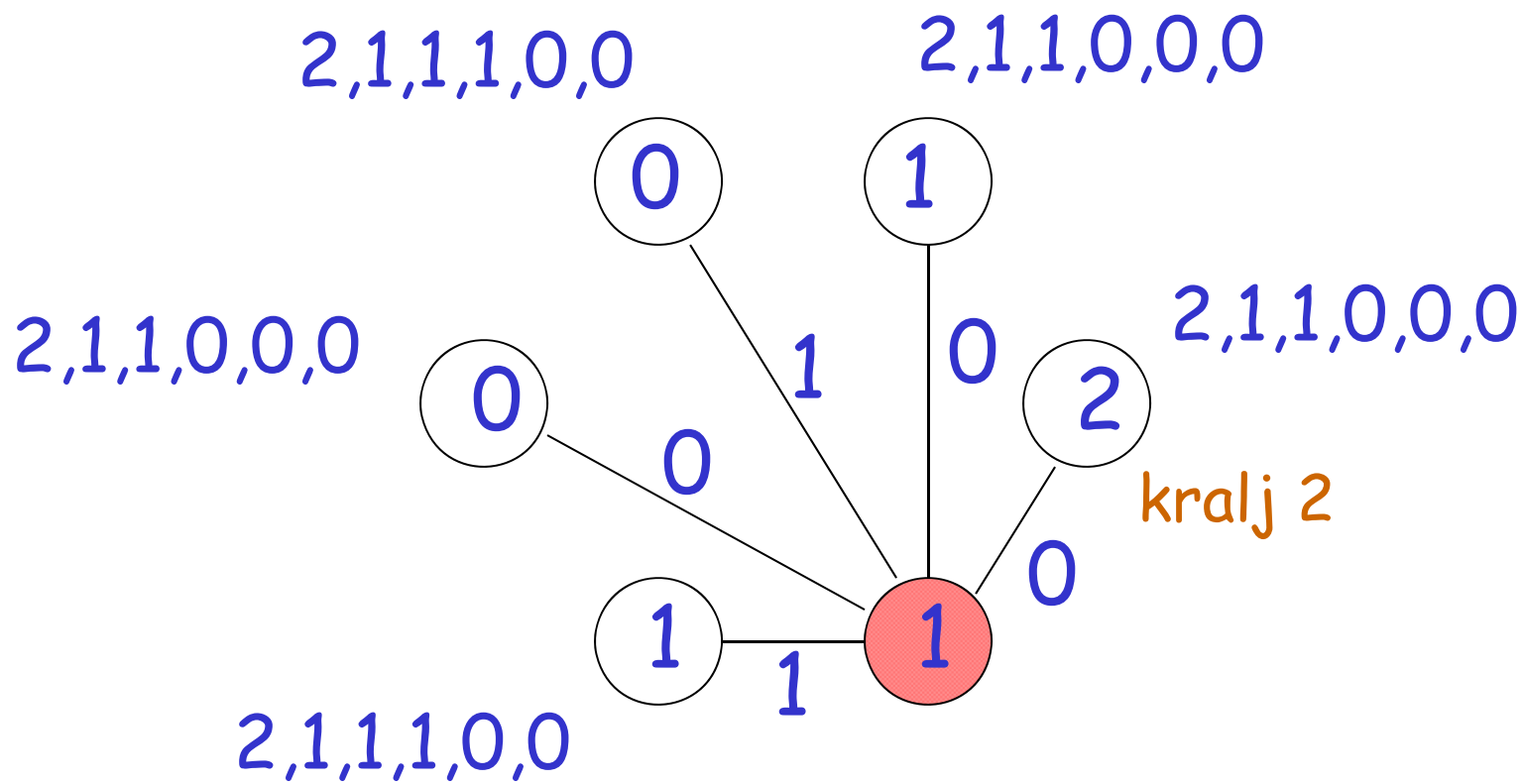
Kralj šalje svima

Faza 1, Runda 2



Svi izabiraju kraljevu vrednost

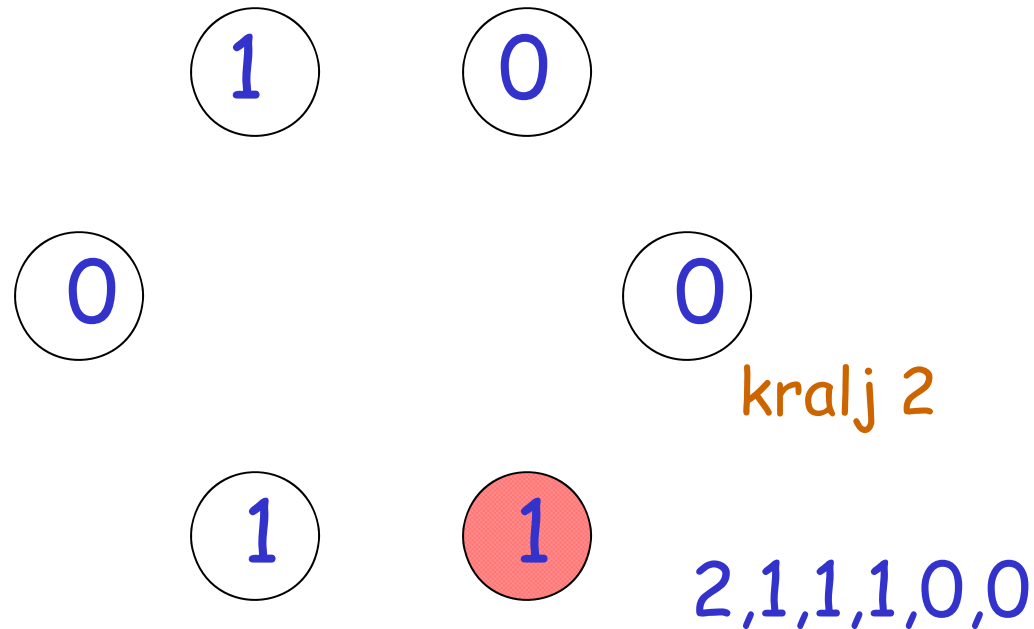
Faza 2, Runda 1



Svi šalju svima

Faza 2, Runda 1

Izaberi većinsku vred

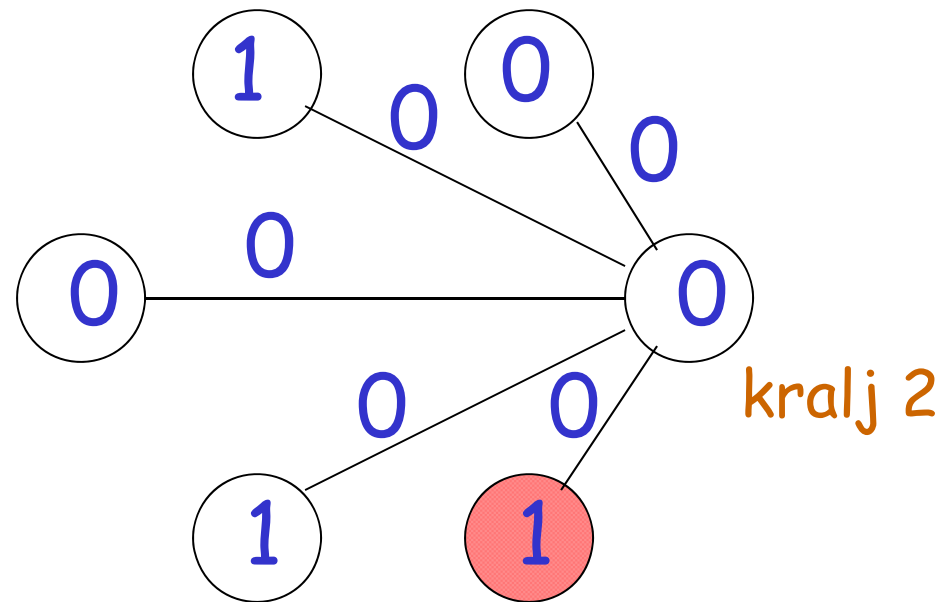


Svima većinski glas bio

$$3 < \frac{n}{2} + f = 4$$

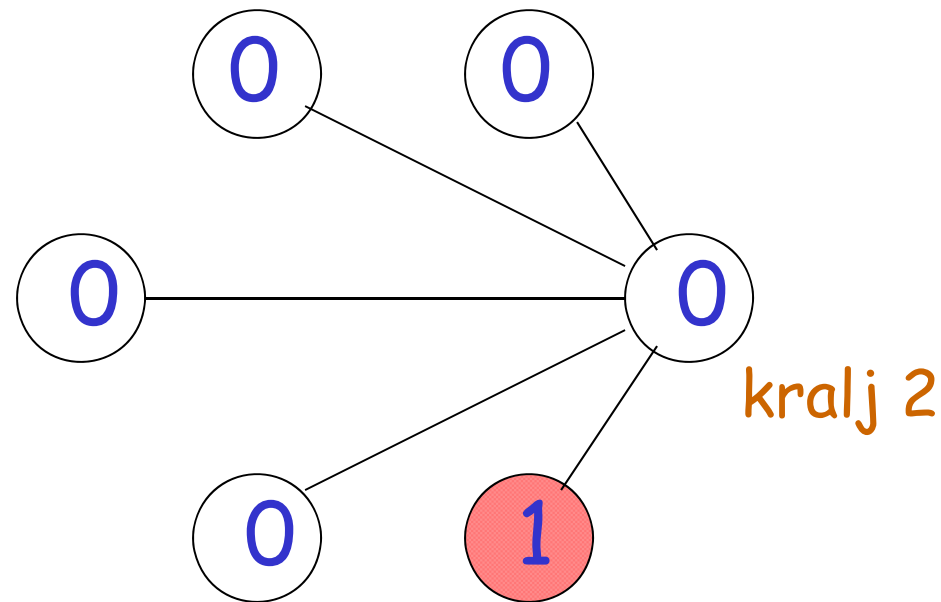
U rundi 2, svi će izabrati kraljevu vrednost

Faza 2, Runda 2



Kralj šalje svima

Faza 2, Runda 2



Svi izabiraju kraljevu vrednost

Konačna odluka

Teorema: U fazi Φ u kojoj
je kralj ispravan,
svi ispravni procesori se
odlučuju za istu vrednost

Dokaz: Razmotrimo fazu Φ

Na kraju runde 1,
ispitujemo dva slučaja:

Sluč. 1: neki čvor je izabrao
svoju prioritetnu vrednost sa
jakom većinom ($> \frac{n}{2} + f$ glasova)

Sluč. 2: ni jedan čvor nije izabrao
svoju prioritetnu vrednost sa
jakom većinom

Sluč. 1: neka je čvor i izabrao svoju prioritetnu vred. a
sa jakim većinom ($> \frac{n}{2} + f$ glasova)

Na kraju runde 1, svi drugi čvorovi
moraju imati prioritetnu vred. a
(uključujući kralja)

Objašnjenje:

Bar $> \frac{n}{2} + f$ ispravnih čvorova je moralo
poslati svima a na početku runde 1

Na kraju runde 2:

Ako čvor zadrži svoju spost. vred.:
onda odlučuje a

Ako čvor dobije vrednost od kralja:
onda on odlučuje a ,
pošto je kralj odlučio a

Zbog toga: Svaki ispravan čvor odlučuje a

Sluč. 2: Ni jedan čvor nije izabrao svoju prior. vred. sa
jakom većinom ($> \frac{n}{2} + f$ glasova)

Svaki ispravan čvor će usvojiti
vrednost od kralja, pa će se svi
odlučiti za istu vrednost

KRAJ DOKAZA

Neka je a vrednost koja je odlučena
na kraju faze Φ

Posle Φ , vred. a će uvek biti prioritarna
sa jakim većinom, pošto je broj
ispravnih procesora: $n - f > \frac{n}{2} + f$

(jer je $f < \frac{n}{4}$)

Zato, od Φ sve do poslednje faze $f + 1$
svaki ispravan procesor odlučuje a

Jedan nemoguć rezultat

Teorema: Ne postoji f -elastičan algoritam
za n procesa, koji zadovoljava

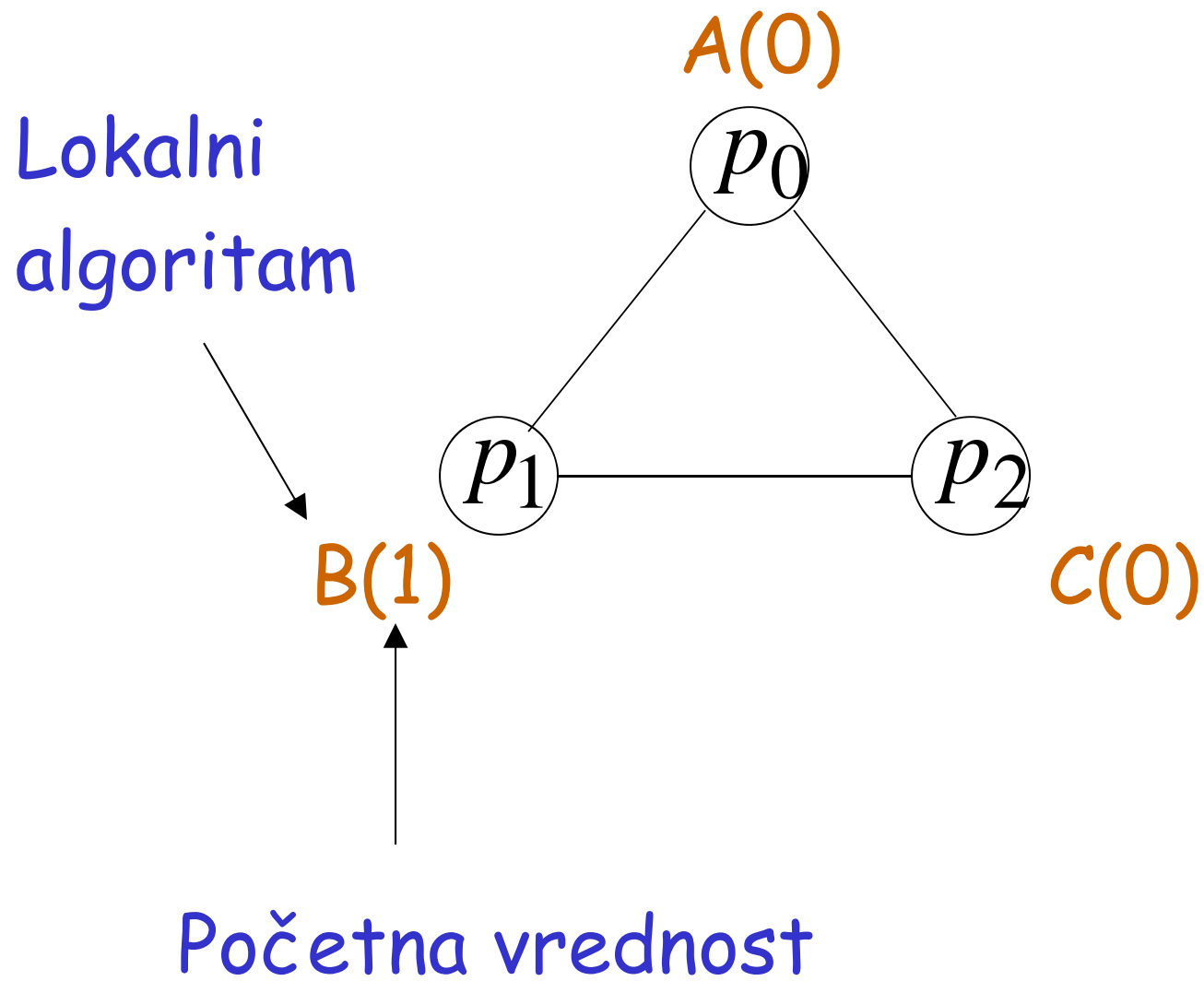
$$f \geq \frac{n}{3}$$

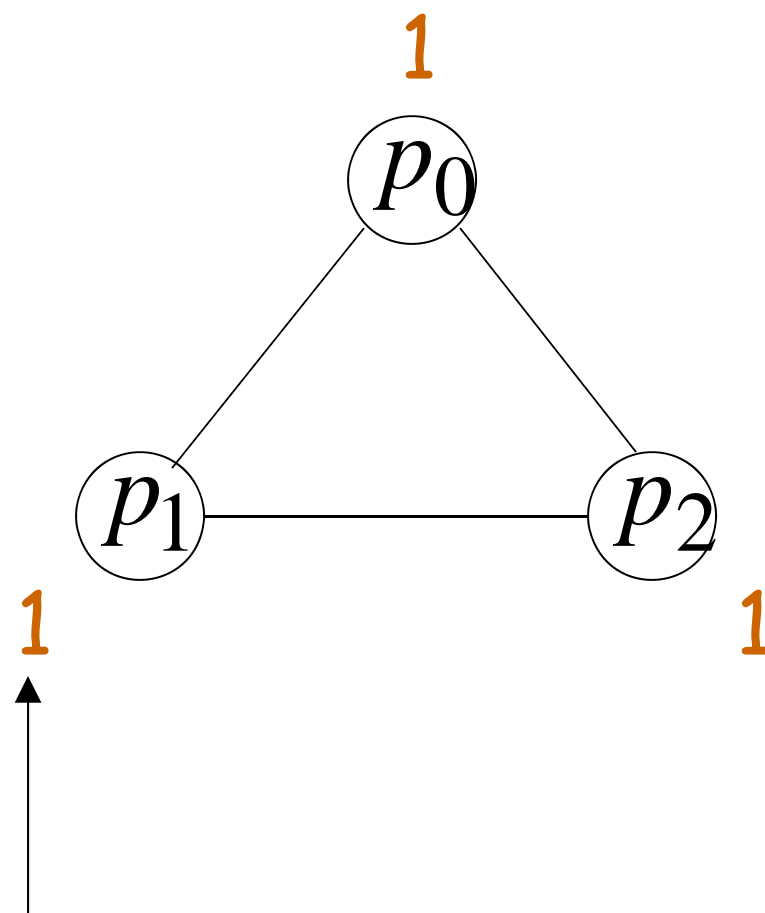
Dokaz: Prvo dokazujemo slučaj sa 3 procesa,
a zatim opšti slučaj

Slučaj sa 3 procesa

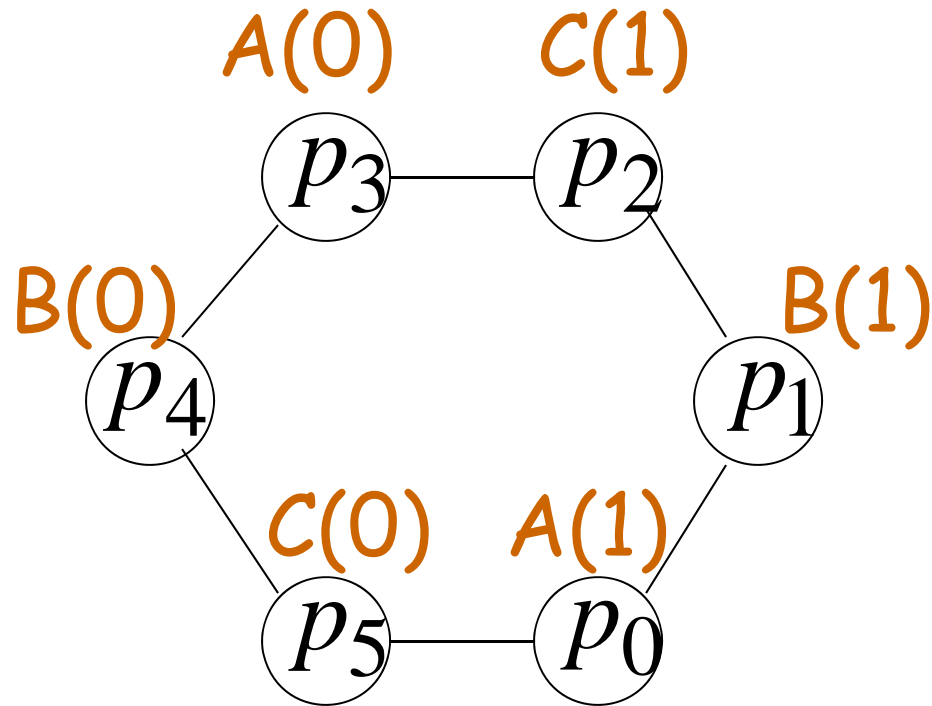
Lema: Ne postoji 1-elastičan algoritam
za 3 procesa

Dokaz: Predpost. radi kontradikcije da postoji
1-elastičan algoritam za 3 procesa



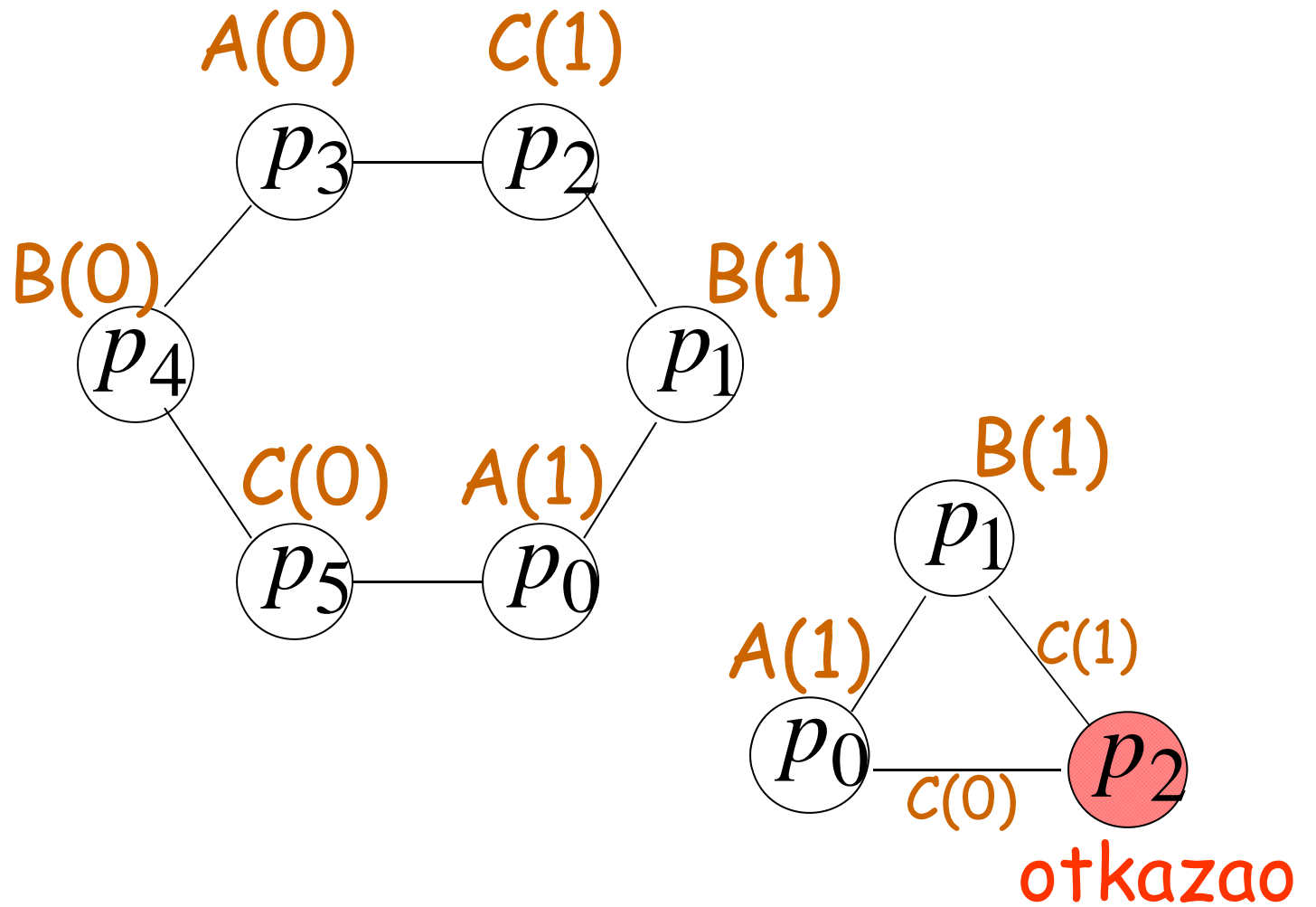


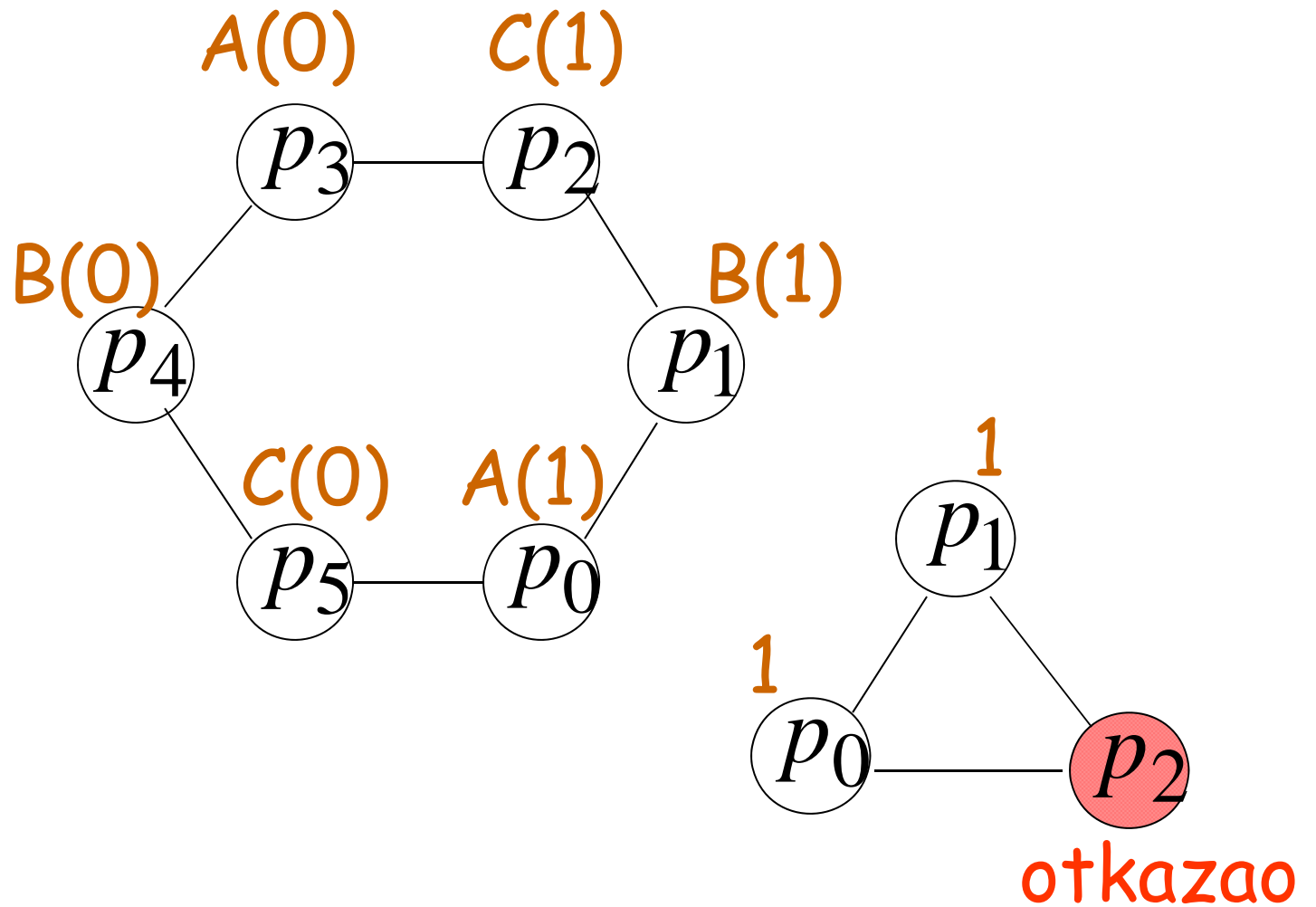
Odlučena vrednost



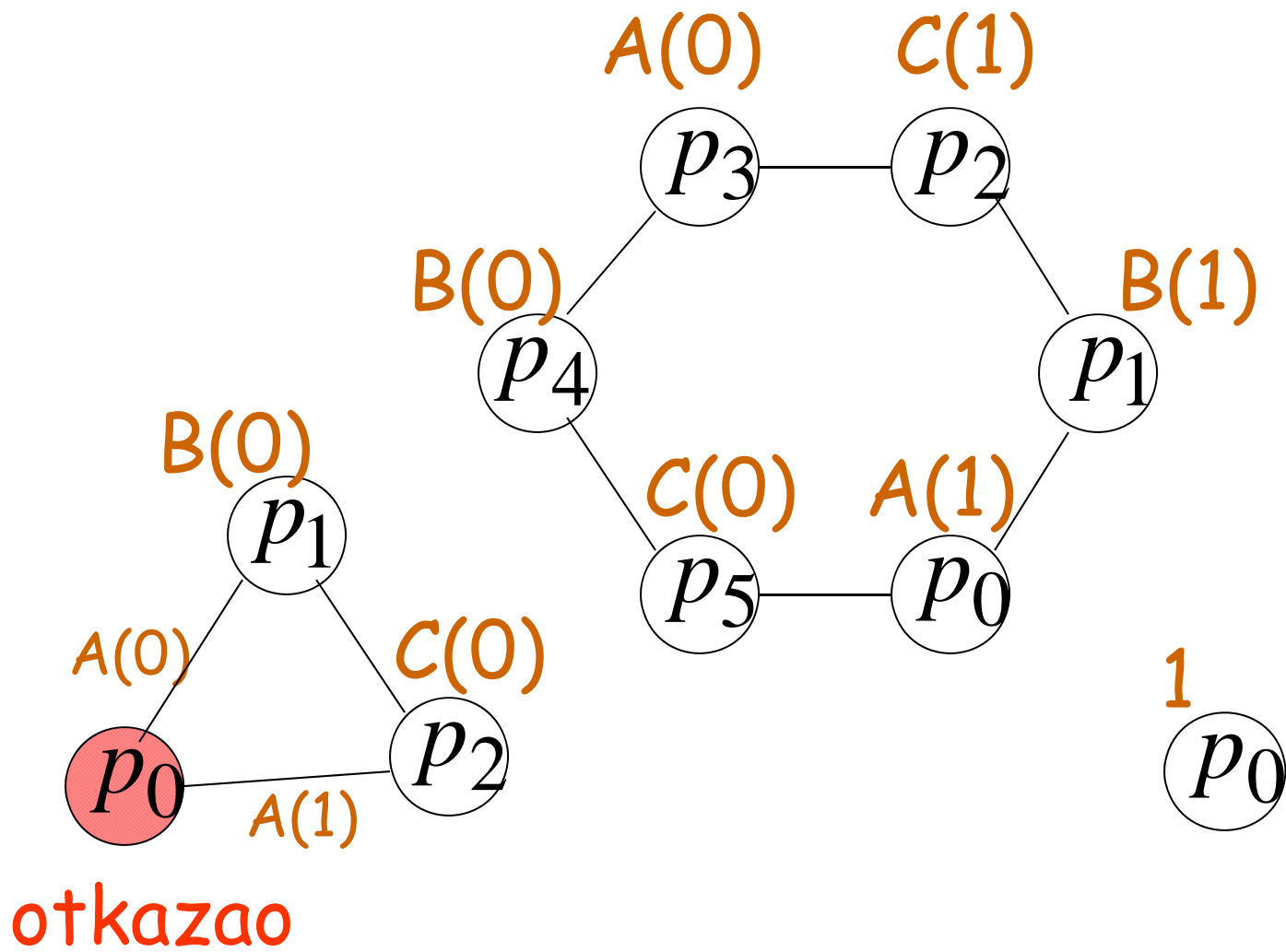
Predpost. da su procesi u prstenu

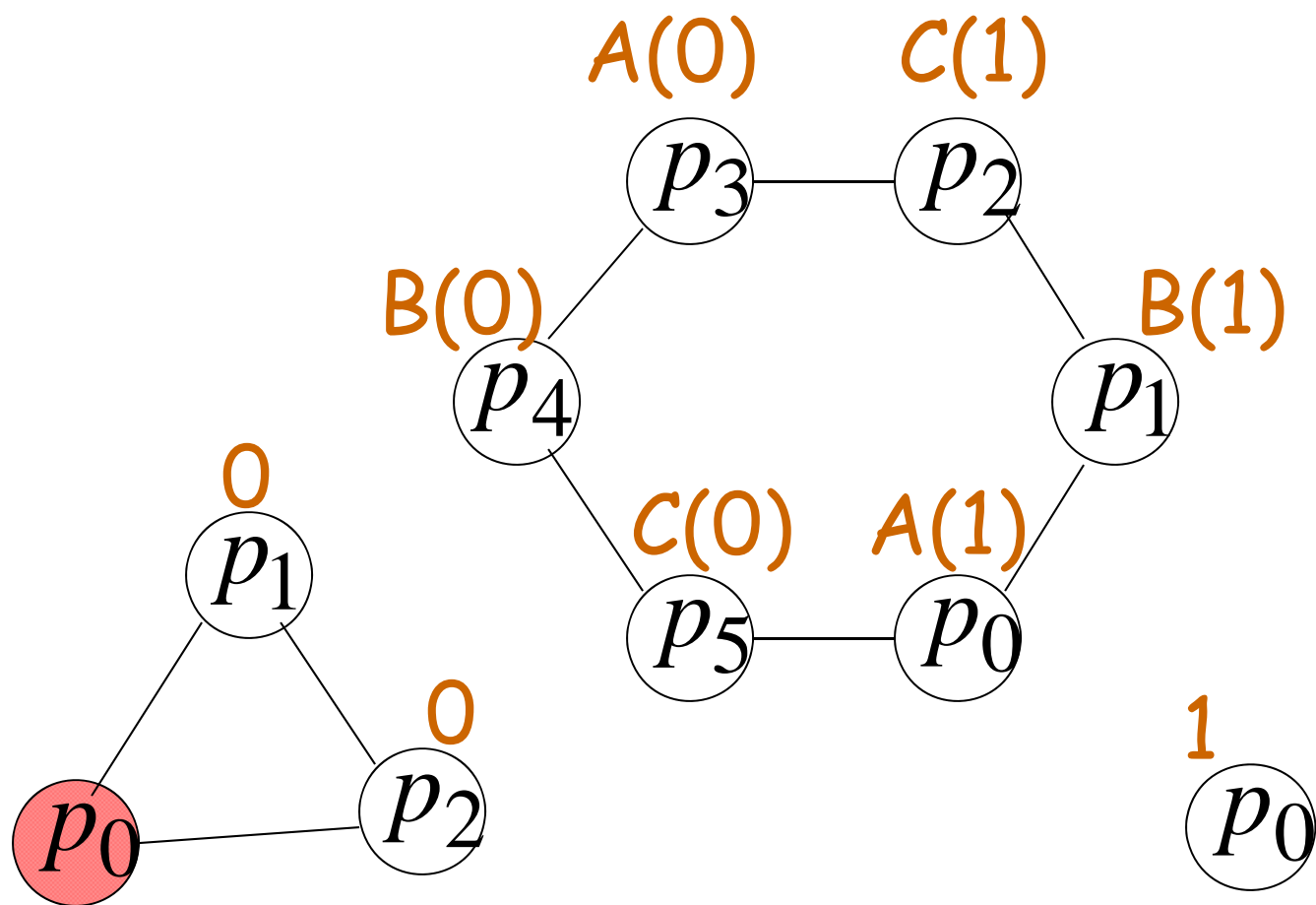
Procesi misle da su u trouglu





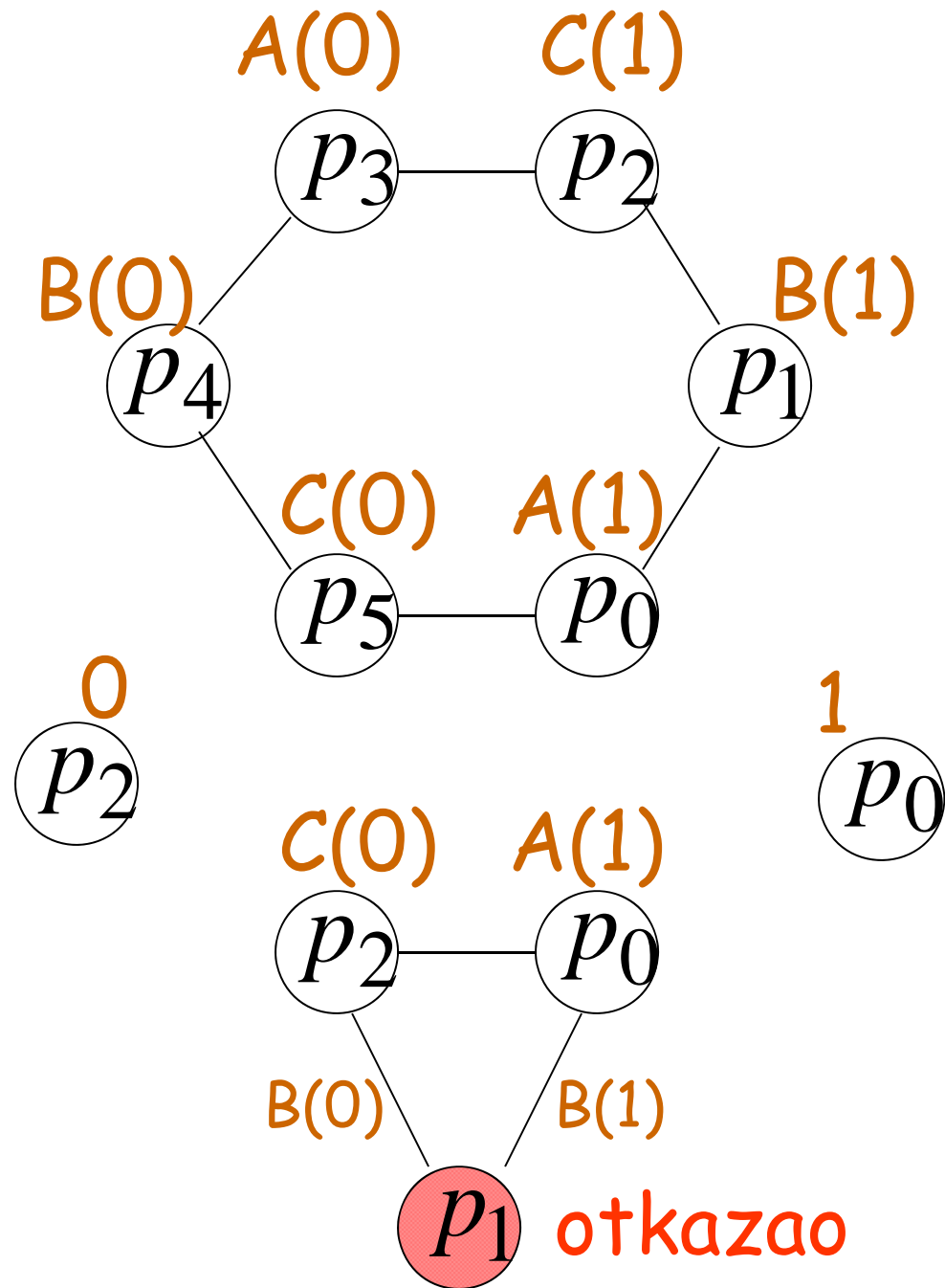
(uslov validnosti)

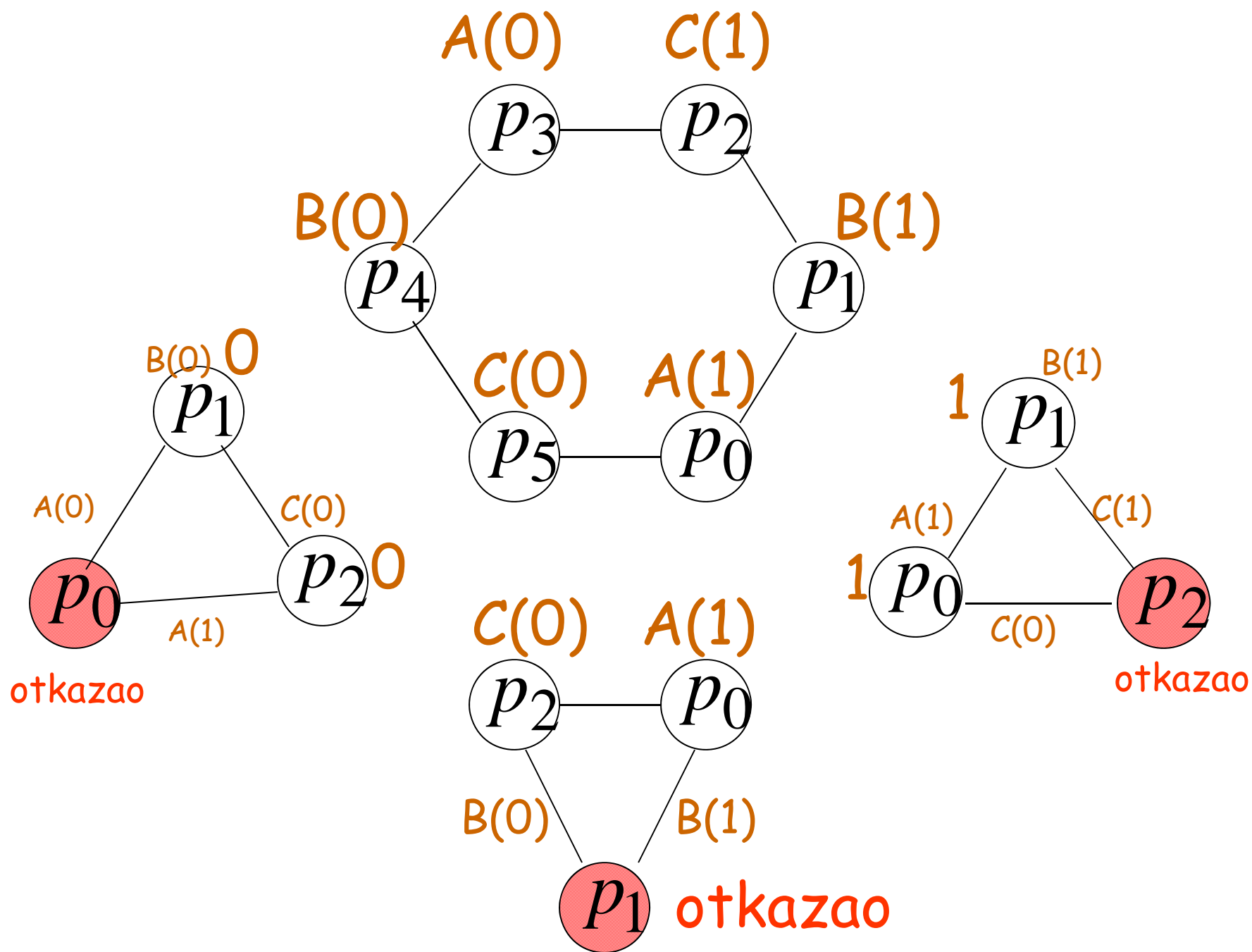


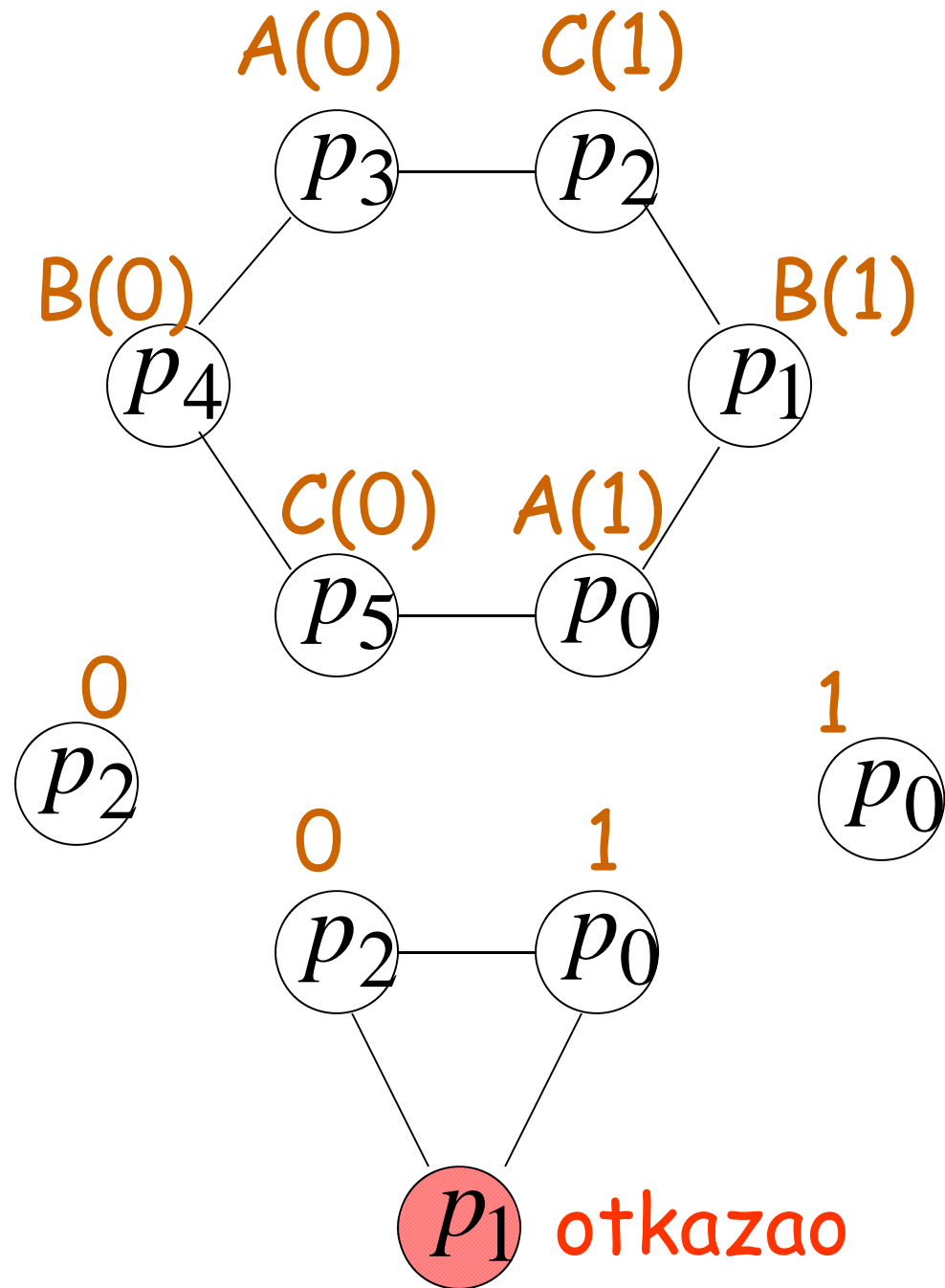


otkazao

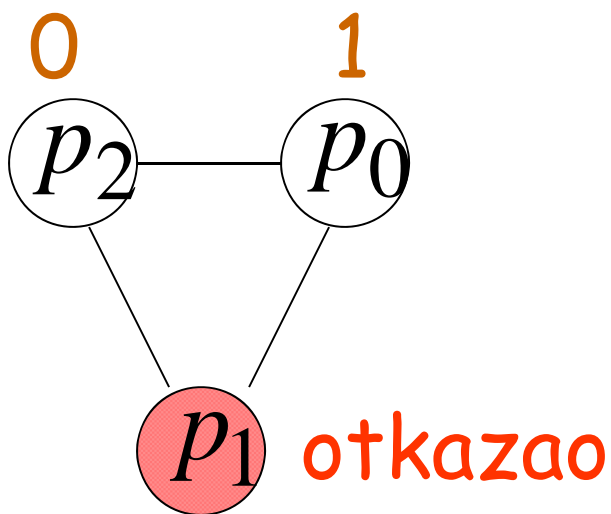
(uslov validnosti)







Nemoguće!!!
jer je algoritam 1-elastičan



Zaključak:

Ne postoji algoritam koji rešava
konsenzus za 3 procesa
od kojih je 1 vizantijski proces

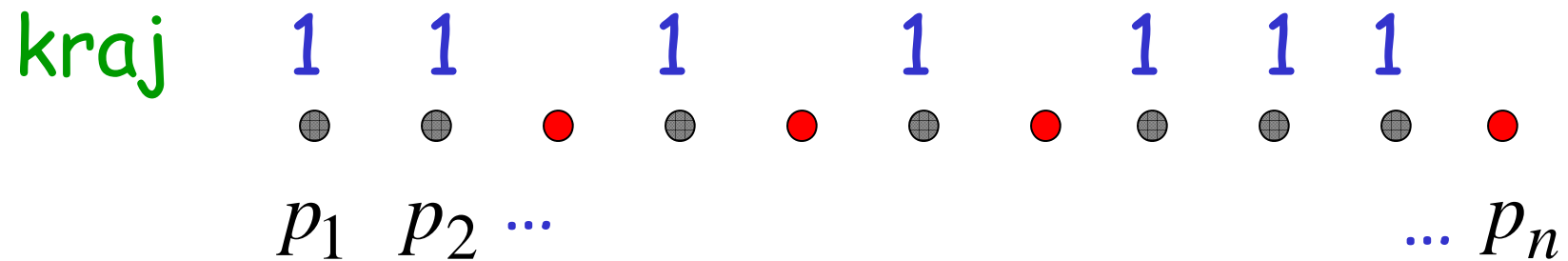
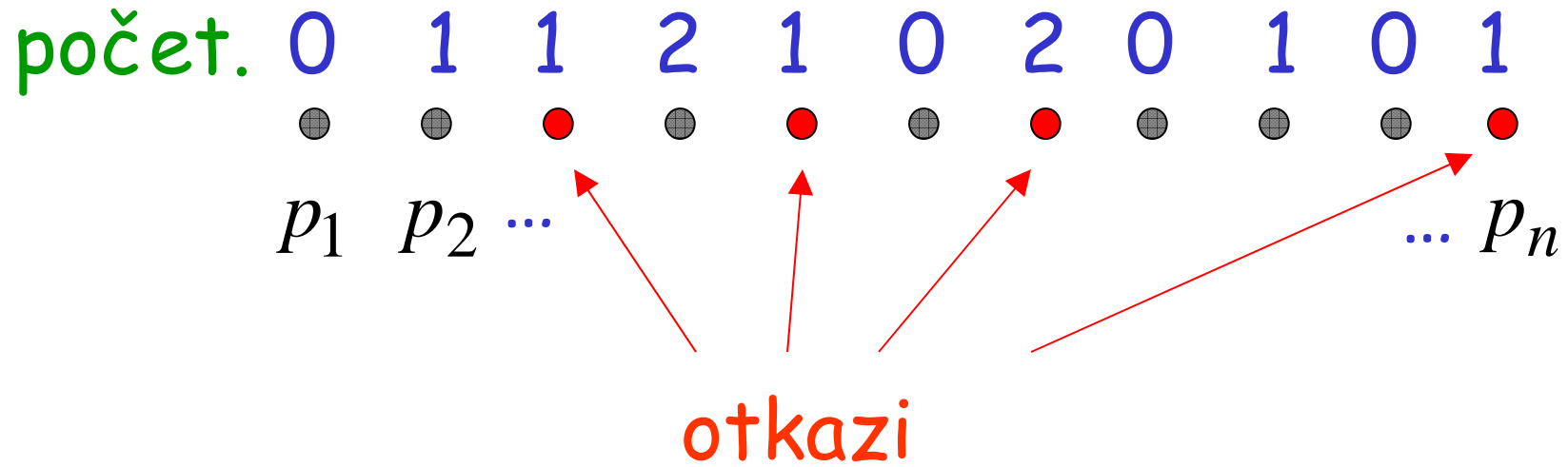
Slučaj sa n procesa

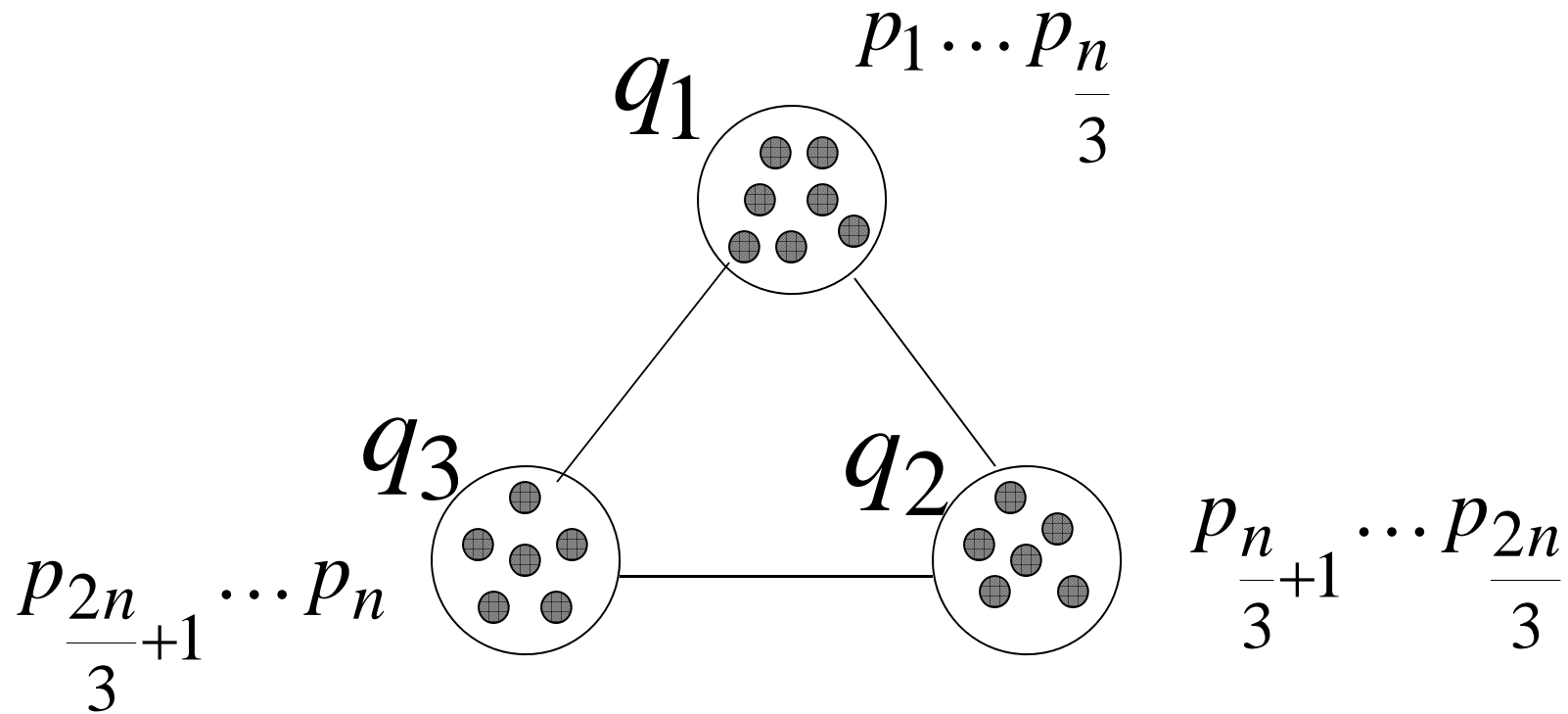
Predpost. radi kontradikcije da postoji neki f -elastičan algoritam A za n procesa, za koji je: $f \geq \frac{n}{3}$

Koristićemo algoritam A da rešimo konsenzus za 3 procesa i 1 otkaz

(kontradikcija)

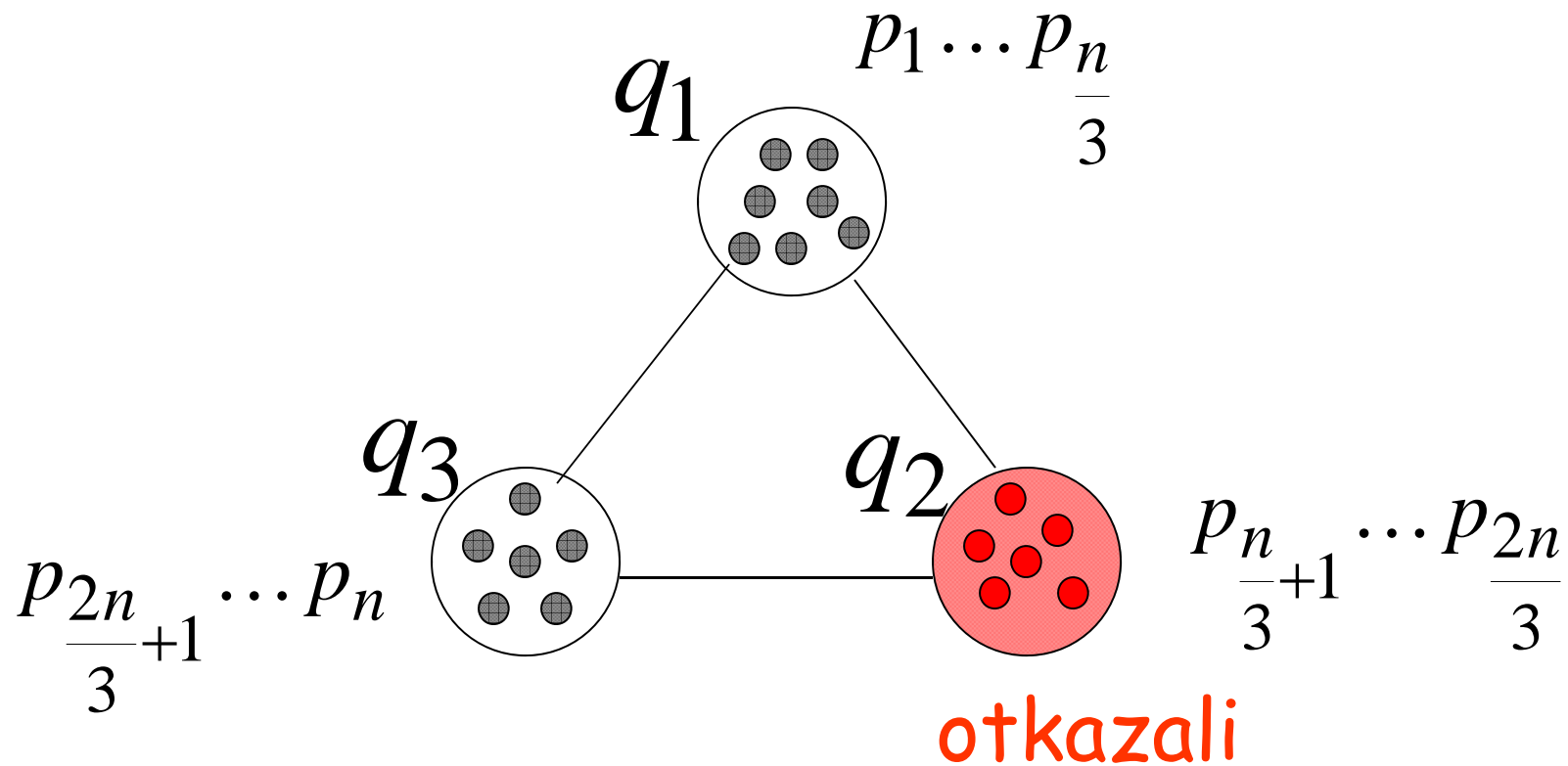
algorithm A





Svaki proces q simulira algoritam A

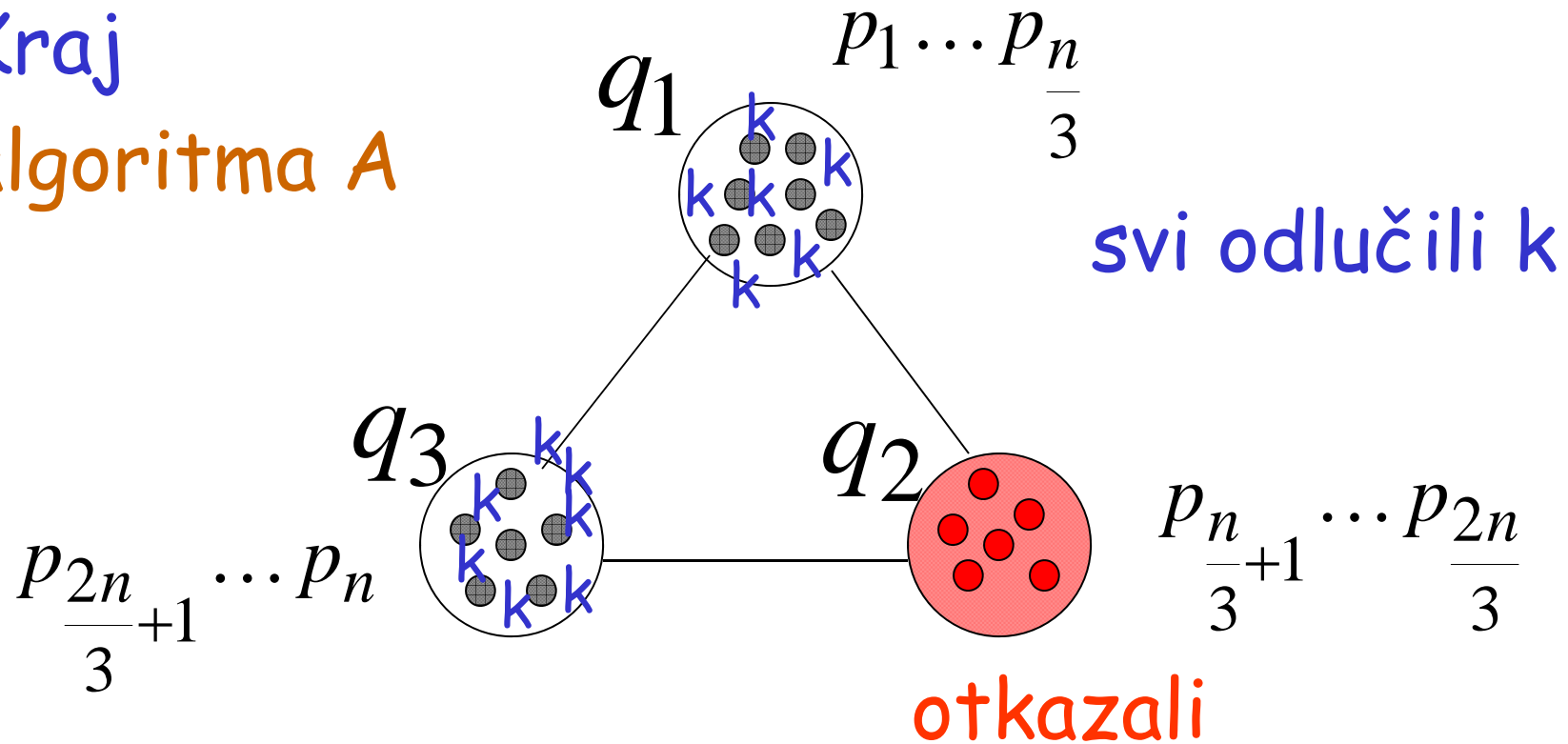
za $\frac{n}{3}$ svih p procesa



Kad q otkaže

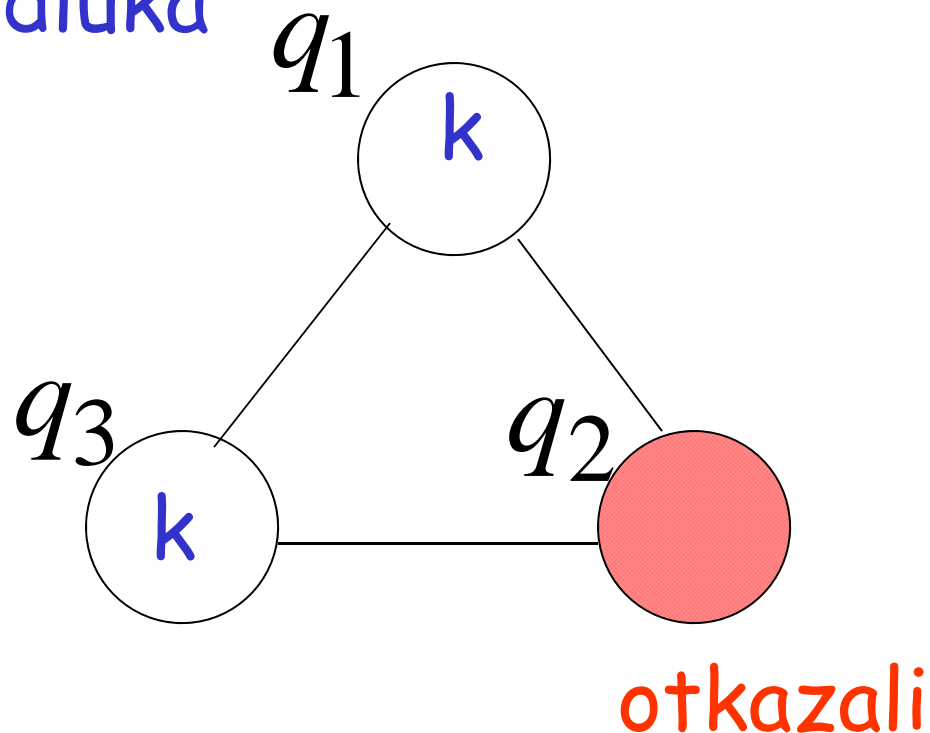
onda $\frac{n}{3}$ svih p procesa takođe otkaže

Kraj
 algoritma A



algoritam A toleriše $\frac{n}{3}$ otkaza

Konačna odluka



Došli smo do konsenzusa sa 1 otkazom

Nemoguće!!!

Zaključak:

Ne postoji f -elastičan algoritam
za n procesa, gde je

$$f \geq \frac{n}{3}$$

Randomizirani vizantijski dogovor

Postoji neki poverljiv procesor q
koji u svakoj rundi baca na slučaj novčić
(coin) i informiše sve druge procesore

coin = heads (verovatnoća $\frac{1}{2}$)

coin = tail (verovatnoća $\frac{1}{2}$)

Svaki procesor P_i ima prioritetsnu vred. v_i

Na početku,
prioritetna vred se postavlja na početnu vred

Predpostavimo da je početna vred. binarna

$$v_i \in \{0,1\}$$

Ovaj algoritam toleriše $f < \frac{n}{8}$
vizantijskih procesora

Postoje tri praga vrednosti:

$$L = \frac{5n}{8} + 1$$

$$H = \frac{6n}{8} + 1$$

$$G = \frac{7n}{8} + 1$$

U svakoj rundi, procesor p_i izvršava:

Šalji svima v_i ;

Primi vrednosti od svih procesora;

$maj_i \leftarrow$ većinska vrednost;

$tally_i \leftarrow$ broj pojava od maj_i ;

If coin=heads then $threshold \leftarrow L = \frac{5n}{8} + 1$

else $threshold \leftarrow H = \frac{6n}{8} + 1$

If $tally_i \geq threshold$ then $v_i \leftarrow maj_i$
else $v_i \leftarrow 0$

If $tally_i \geq G = \frac{7n}{8} + 1$ then došlo se do odluke

Analiza: Ispitajmo slučajeve u rundi

Završetak: Postoji neki procesor p_i
sa $tally_i \geq G = \frac{7n}{8} + 1$

Drugi slučajeve:

Sluč. 1: Dva procesora p_i i p_k imaju
različite $maj_i \neq maj_k$

Sluč. 2: Svi procesori imaju isti maj_i

Završetak: Postoji neki procesor p_i
sa $tally_i \geq G = \frac{7n}{8} + 1$

Pošto procesora u otkazu ima najviše $f < \frac{n}{8}$

procesor p_i prima bar

$$tally_i - f \geq \frac{6n}{8} + 1$$

glasova za maj_i od dobrih procesora

Zbog toga, svaki procesor p_k

će imati $maj_i = maj_k$

sa $tally_k \geq H = \frac{6n}{8} + 1$

Sledstveno, na kraju runde
svi dobri procesori će imati istu
prioritetnu vrednost:

$$v_k = maj_k = maj_i$$

Opažanje:

Ako na početku runde svi dobri procesori imaju istu prioritetnu vrednost onda se algoritam završava u toj rundi

Ovo važi jer će za svaki procesor p_i uslov završetka $tally_i \geq G = \frac{7n}{8} + 1$ biti zadovoljen u toj rundi

Zbog toga, ako je uslov završetka zadovoljen za jedan procesor u nekoj rundi, onda, će uslov završetka biti zadovoljen za sve procesore u sledećoj rundi.

Sluč. 1: Dva procesora p_i i p_k imaju različite $maj_i \neq maj_k$

Mora biti da je $tally_i < L = \frac{5n}{8} + 1$

i da je $tally_k < L = \frac{5n}{8} + 1$

I zbog toga je $v_i = v_k = 0$

Zato, svaki procesor bira 0,
i algoritam se završava u sledećoj rundi

Predpost. (radi kontradikcije) da je

$$\mathit{tally}_i \geq L = \frac{5n}{8} + 1$$

Onda je bar

$$\mathit{tally}_i - f \geq \frac{4n}{8} + 1 = \frac{n}{2} + 1$$

dobrih procesora glasalo maj_i

Sledstveno, $\mathit{maj}_i = \mathit{maj}_j$

Kontradikcija!

Sluč. 2: Svi procesori imaju isti maj_i

Onda za bilo koja dva procesora p_i i p_k
važi da je $|tally_i - tally_k| \leq f$

jer bi inače, broj procesora
u otkazu bio veći od f

Neka je p_{\min} procesor sa

$$tally_{\min} = \min_i \{tally_i\}$$

Pod-sluč. 1: $tally_{\min} < L = \frac{5n}{8} + 1$

Ako je $threshold = H = \frac{6n}{8} + 1$

(ovo se dešava sa verovatnoćom $\frac{1}{2}$)

onda, za bilo koji procesor p_k važi da je

$$tally_k \leq tally_{\min} + f < L + f = \frac{6n}{8} + 1 = H$$

I zbog toga je $v_i = v_k = 0$

Dakle, svaki procesor izabira 0,
i algoritam se završava u sledećoj rundi

(ovo se dešava sa verovatnoćom $\frac{1}{2}$)

Pod-sluč. 2: $tally_{\min} \geq L = \frac{5n}{8} + 1$

Ako je $threshold = L = \frac{5n}{8} + 1$

(ovo se dešava sa verovatnoćom $\frac{1}{2}$)

onda, za bilo koji procesor p_k važi da je

$$tally_k \geq tally_{\min} \geq L$$

I zbog toga je $V_k = V_{\min}$

Dakle, svaki procesor izabira V_{\min} ,
i algoritam se završava u sledećoj rundi

(ovo se dešava sa verovatnoćom $\frac{1}{2}$)