

# DISTRIBUIRANI ALGORITMI I SISTEMI

# Konsenzus sa vizantijskim otkazima

2

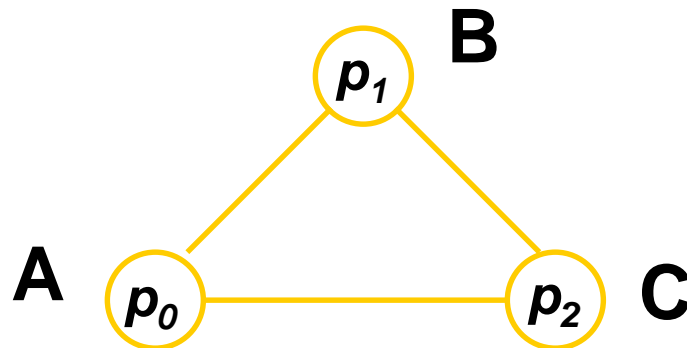
- Koliko procesora je potrebno da se reši konsenzus kad je  $f = 1$  ?
- Predpost.  $n = 2$ . Ako  $p_0$  ima ulaz 0 a  $p_1$  ima 1, jedan mora da promeni vred, ali ne oba. Šta ako je jedan proc u otkazu? Kako drugi to može znati?
- Predpost.  $n = 3$ . ako  $p_0$  ima ulaz 0,  $p_1$  ima ulaz 1, i  $p_2$  je u otkazu, onda je potreban arbitar, ali  $p_2$  može delovati zlonamerno

# Donja granica za procesore za $f = 1$

3

**Teorema (10.7):** Svaki algoritam konsenzusa za 1 vizantijski otkaz mora imati bar 4 procesora

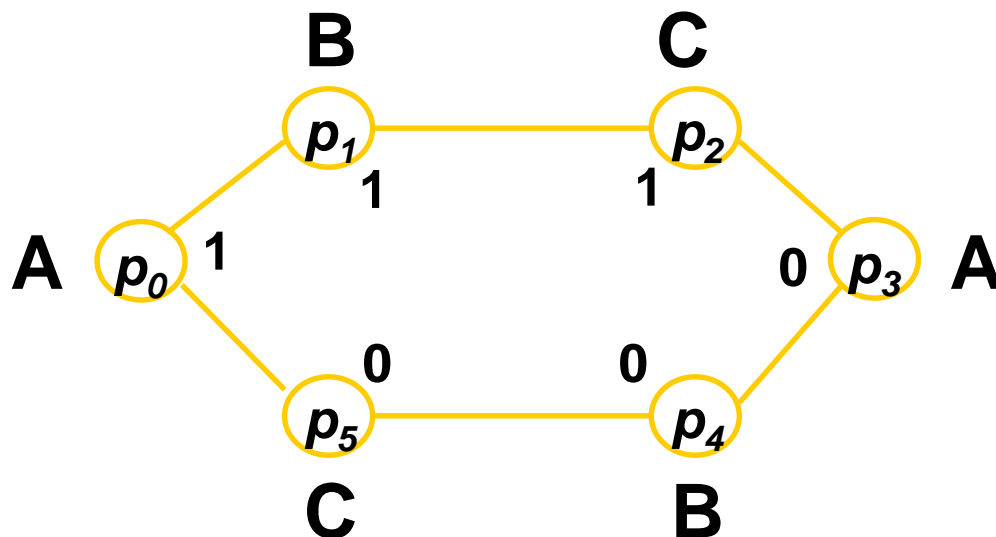
**Dokaz:** Predpost. radi kontradikcije da postoji algoritam konsenzusa  $\mathcal{A} = (A, B, C)$  za 3 procesora i 1 vizantijski otkaz



# Specificiranje ponašanja otkaza

4

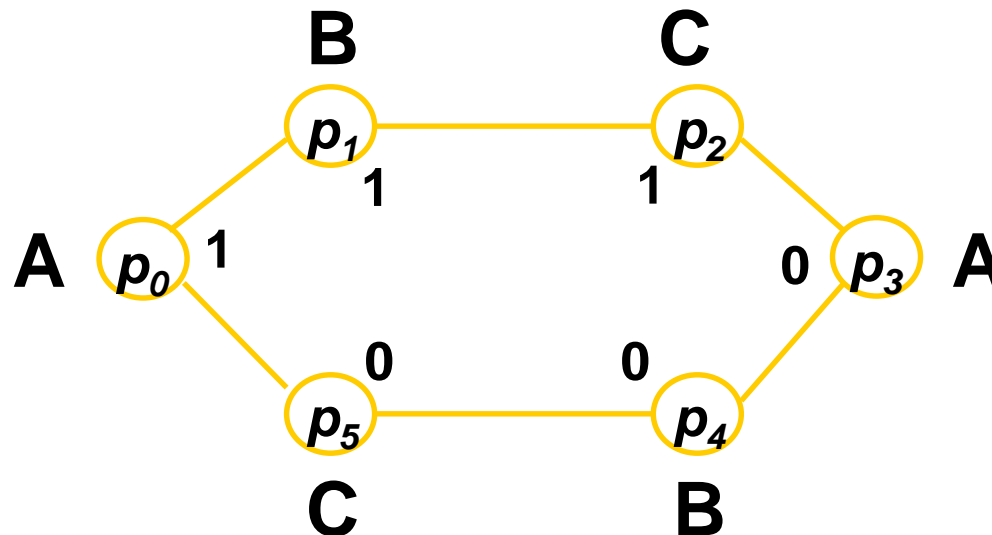
- Razmotrimo prsten sa 6 ispravnih procesora koji izvršavaju komponente od  $\mathcal{A}$  na ovaj način:



# Specificiranje ponašanja otkaza

4

- Razmotrimo prsten sa 6 ispravnih procesora koji izvršavaju komponente od  $\mathcal{A}$  na ovaj način:

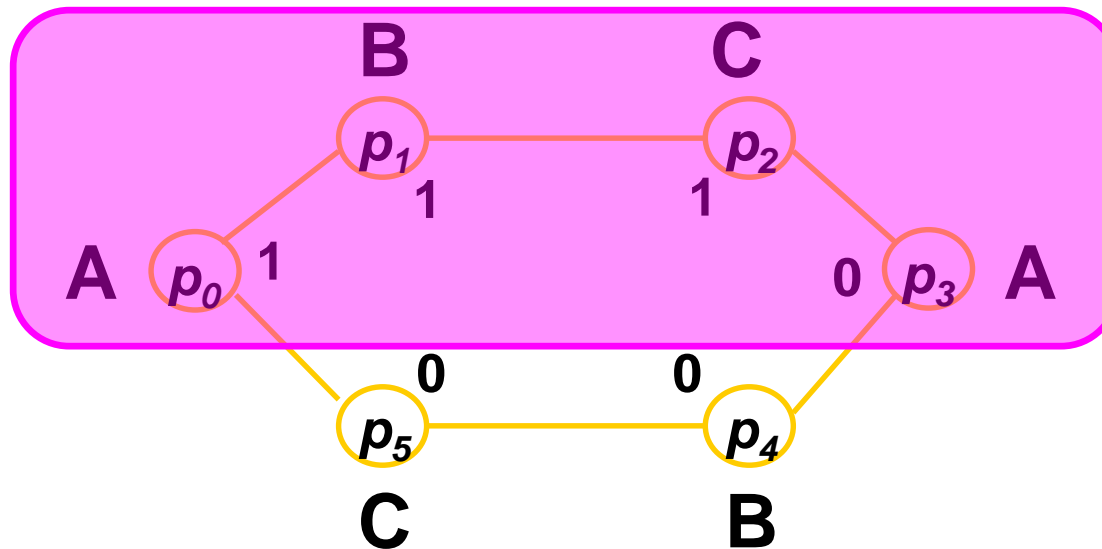


- Ovo izvršenje  $\beta$  možda (ne) rešava konsenzus (ne mora da uspe)
- Ali procesori rade *nešto* – to ponašanje se koristi za specifikaciju ponašanja procesora u otkazu u izvršenju  $\mathcal{A}$  u trouglu

# Specificiranje ponašanja otkaza

4

- Razmotrimo prsten sa 6 ispravnih procesora koji izvršavaju komponente od  $\mathcal{A}$  na ovaj način:

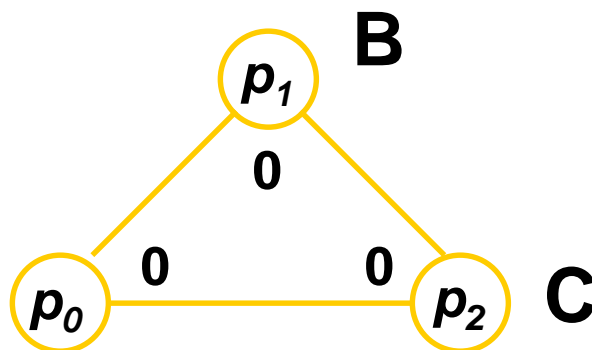


- Ovo izvršenje  $\beta$  možda (ne) rešava konsenzus (ne mora da uspe)
- Ali procesori rade *nešto* – to ponašanje se koristi za specifikaciju ponašanja procesora u otkazu u izvršenju  $\mathcal{A}$  u trouglu

# Priprema kontradikcije

5

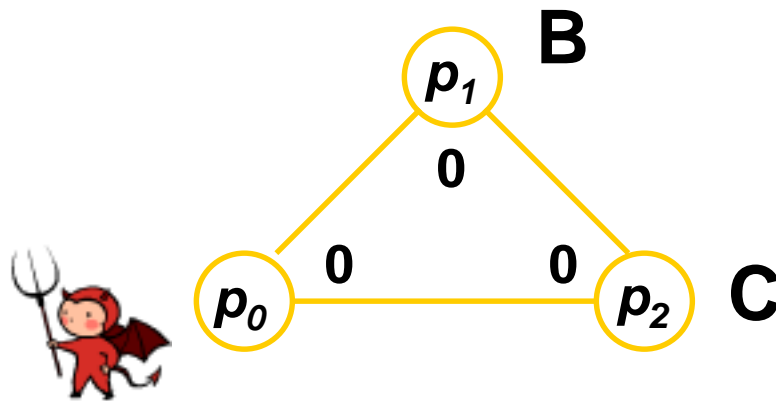
□ Neka je  $\alpha_0$  ovo izvršenje:



# Priprema kontradikcije

5

- Neka je  $\alpha_0$  ovo izvršenje:

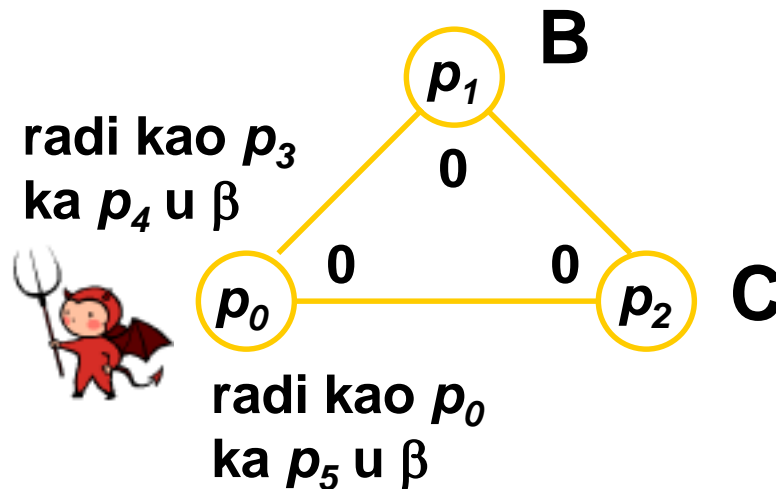




# Priprema kontradikcije

5

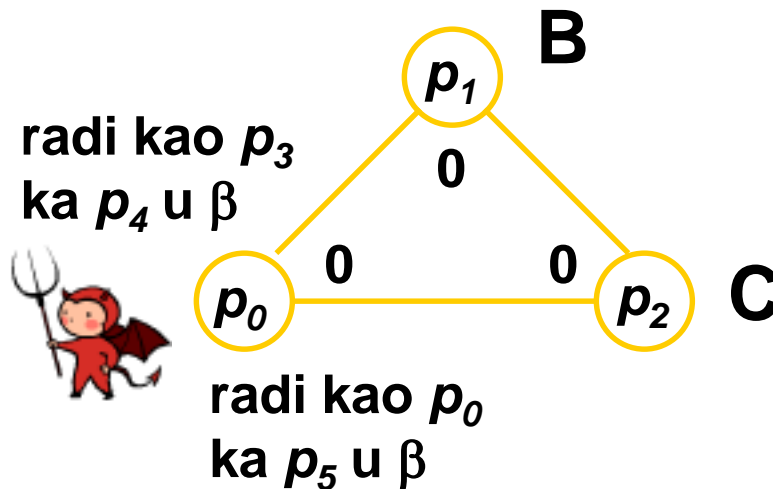
- Neka je  $\alpha_0$  ovo izvršenje:



# Priprema kontradikcije

5

- Neka je  $\alpha_0$  ovo izvršenje:

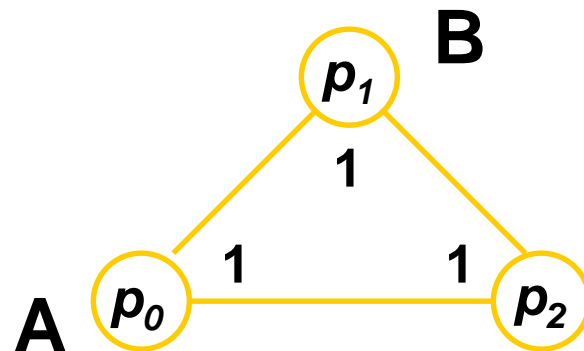


$p_1$  i  $p_2$  moraju  
da odluče 0

# Priprema kontradikcije

6

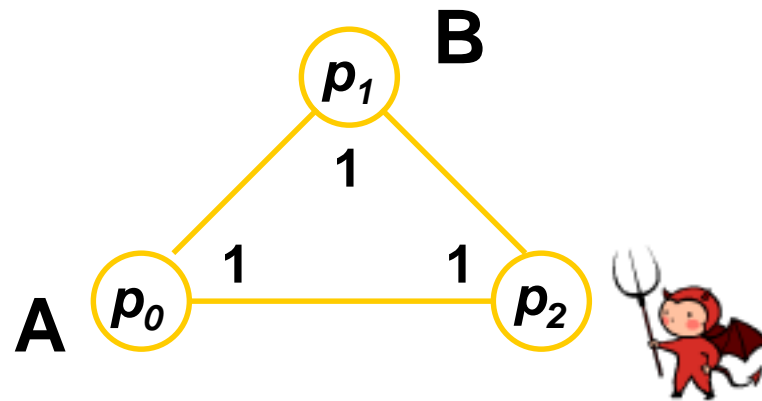
□ Neka je  $\alpha_1$  ovo izvršenje:



# Priprema kontradikcije

6

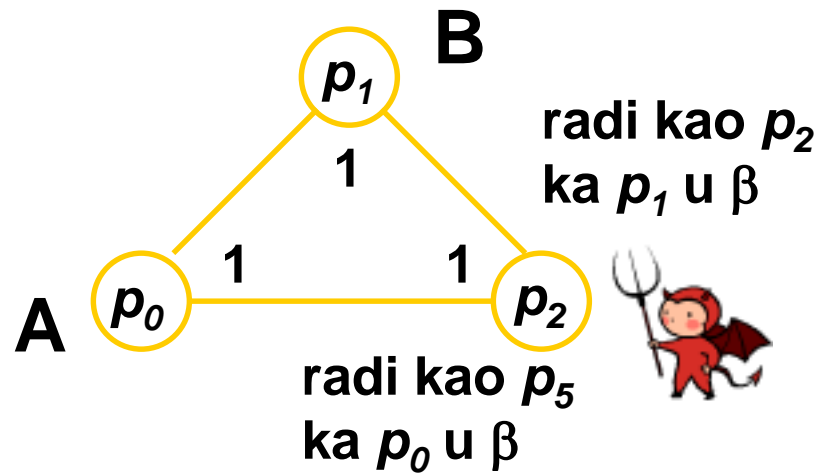
- Neka je  $\alpha_1$  ovo izvršenje:



# Priprema kontradikcije

6

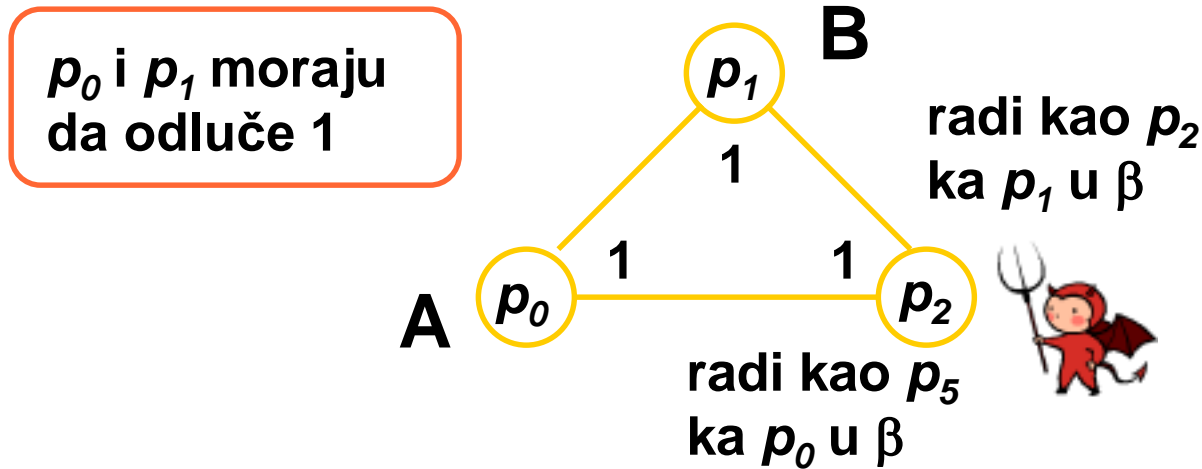
- Neka je  $\alpha_1$  ovo izvršenje:



# Priprema kontradikcije

6

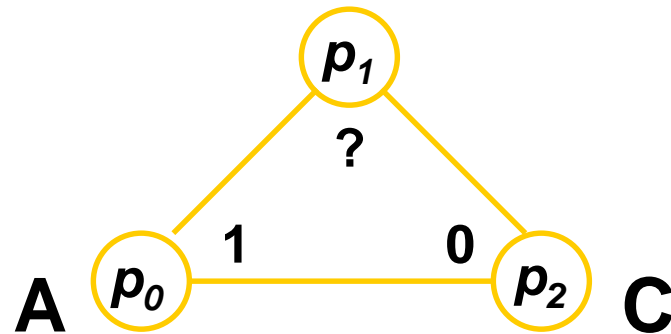
□ Neka je  $\alpha_1$  ovo izvršenje:



# Kontradikcija

7

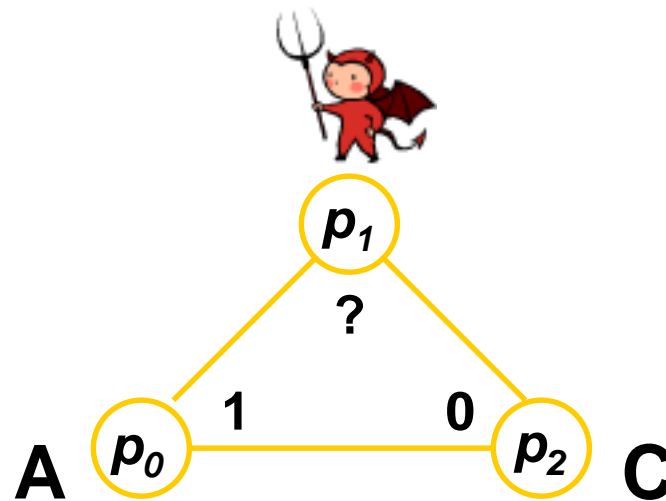
□ Neka je  $\gamma$  ovo izvršenje:



# Kontradikcija

7

- Neka je  $\gamma$  ovo izvršenje:

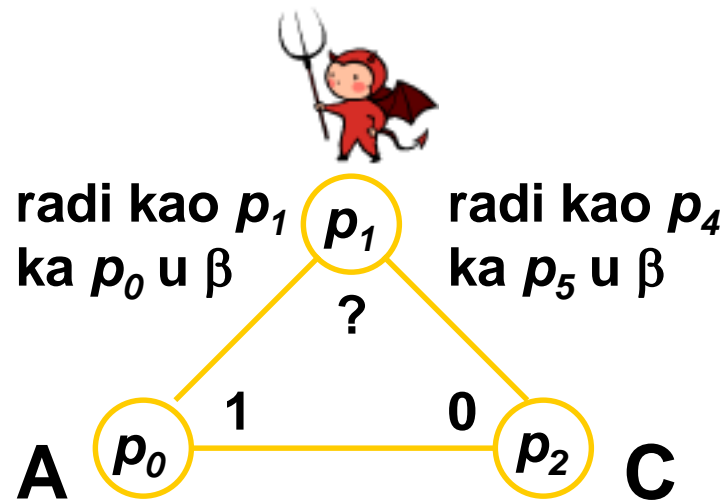




# Kontradikcija

7

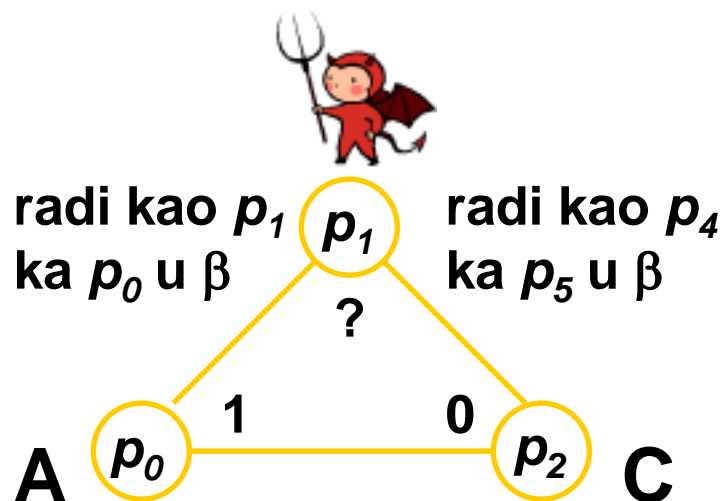
- Neka je  $\gamma$  ovo izvršenje:



# Kontradikcija

7

- Neka je  $\gamma$  ovo izvršenje:

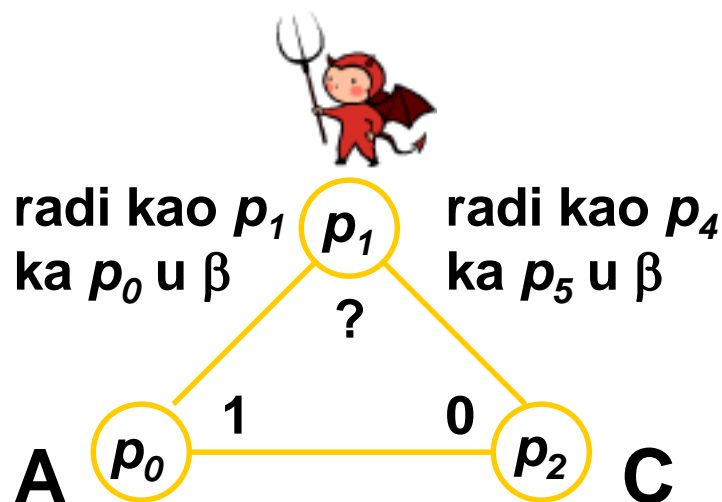


Šta  $p_0$  i  $p_2$   
odlučuju?

# Kontradikcija

7

- Neka je  $\gamma$  ovo izvršenje:



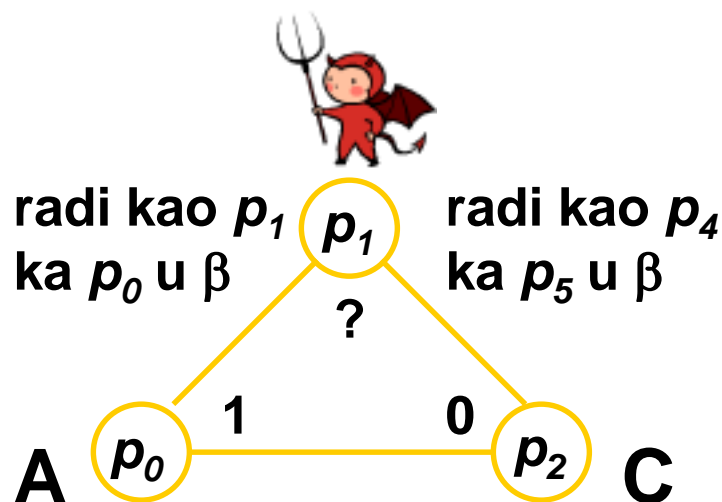
Šta  $p_0$  i  $p_2$   
odlučuju?

pogled  $p_0$  u  $\gamma$  = pogled  $p_0$  u  $\beta$  = pogled  $p_0$  u  $\alpha_1 \Rightarrow p_0$  odlučuje 1

# Kontradikcija

7

- Neka je  $\gamma$  ovo izvršenje:



Šta  $p_0$  i  $p_2$   
odlučuju?

pogled  $p_0$  u  $\gamma$  = pogled  $p_0$  u  $\beta$  = pogled  $p_0$  u  $\alpha_1 \Rightarrow p_0$  odlučuje 1

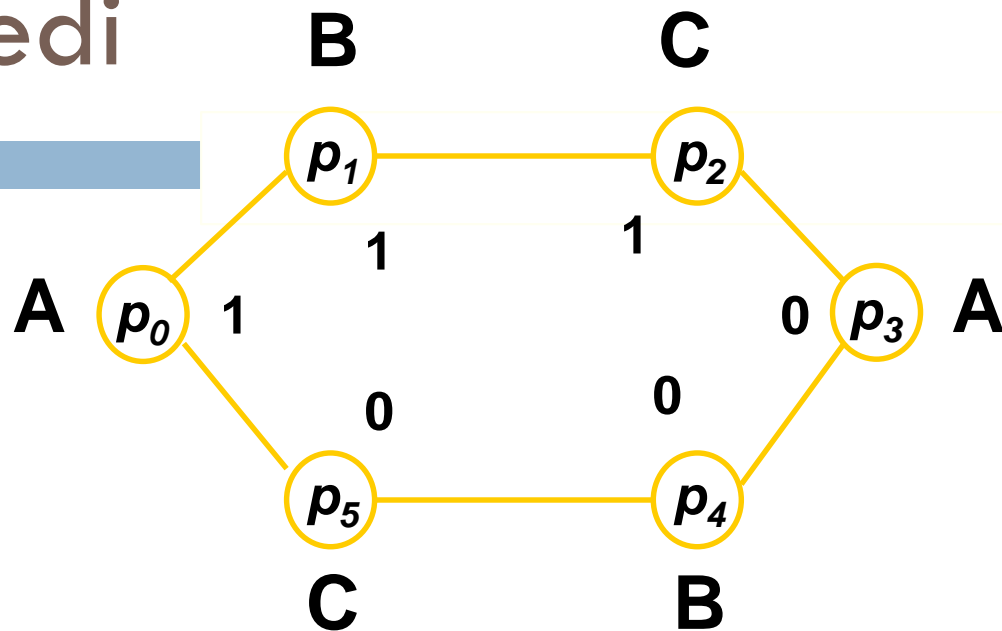
pogled  $p_2$  u  $\gamma$  = pogled  $p_5$  u  $\beta$  = pogled  $p_2$  u  $\alpha_0 \Rightarrow p_2$  odlučuje 0

**Kontradikcija!**

# Pogledi

8

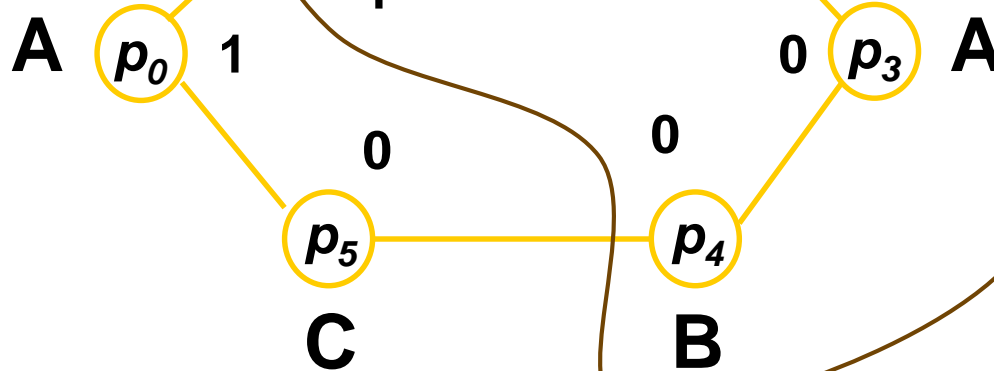
$\beta$ :



# Pogledi

8

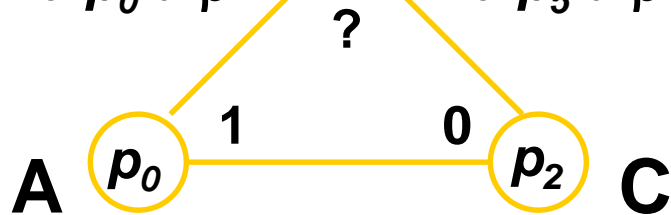
$\beta$ :



$\gamma$ :



radi kao  $p_1$  ka  $p_0$  u  $\beta$  radi kao  $p_4$  ka  $p_5$  u  $\beta$



# Pogledi

8

$\beta$ :

A

$p_0$

1

$p_1$

1

$p_5$

0

C

C

$p_2$

1

$p_4$

0

B

$p_3$

0

A

$\gamma$ :



radi kao  $p_1$   
ka  $p_0$  u  $\beta$

$p_1$

?

radi kao  $p_4$   
ka  $p_5$  u  $\beta$

A

$p_0$

1

$p_2$

0

C

# Pogledi

8

$\beta$ :

A

$p_0$

1

$p_1$

1

$p_2$

1

$p_3$

0

A

$p_5$

0

$p_4$

0

C

B

$\gamma$ :



radi kao  $p_1$   
ka  $p_0$  u  $\beta$

$p_1$

radi kao  $p_4$   
ka  $p_5$  u  $\beta$

?

1

0

A

$p_0$

$p_2$

C

$\alpha_1$ :

$p_1$

B

1

radi kao  $p_2$   
ka  $p_1$  u  $\beta$

1

1

A

$p_0$

$p_2$

radi kao  $p_5$   
ka  $p_0$  u  $\beta$





# Pogledi

8

$\beta$ :

A

$p_0$

1

$p_1$

1

$p_5$

0

$p_2$

1

$p_3$

0

$p_4$

0

C

B

A

$\gamma$ :



radi kao  $p_1$   
ka  $p_0$  u  $\beta$

$p_1$

?

radi kao  $p_4$   
ka  $p_5$  u  $\beta$

1

0

A

$p_0$

$p_2$

C

$\alpha_1$ :

$p_1$

B

1

radi kao  $p_2$   
ka  $p_1$  u  $\beta$

1

1

A

$p_0$

$p_2$

radi kao  $p_5$   
ka  $p_0$  u  $\beta$



# Donja granica broja procesora za bilo koje $f$

9

**Teorema 10.8:** Bilo koji algoritam konsenzusa za  $f$  vizantijskih otkaza mora imati bar  $3f+1$  procesora

**Dokaz:** Koristi redukciju na 3:1 slučaj

- Predpost. radi kontradikcije da postoji algoritam  $\mathcal{A}$  za  $f > 1$  otkaza i  $n = 3f$  procesora
- Koristimo  $\mathcal{A}$  kao podprogram za konstruisanje algoritma za 1 otkaz i 3 procesora, što dovodi do kontradikcije

# Redukcija

10

- Podelimo  $n \leq 3f$  procesora u 3 skupa,  $P_0$ ,  $P_1$ , i  $P_2$ , svaki veličine najviše  $f$
- U slučaju  $n = 3$ , neka procesori:
  - $p_0$  simulira  $P_0$
  - $p_1$  simulira  $P_1$
  - $p_2$  simulira  $P_2$
- Ako je 1 procesor u otkazu u sistemu sa  $n = 3$ , onda je najviše  $f$  procesora u otkazu u simuliranom sistemu
- Zato je simulirani sistem korektan
- Neka procesori u sistemu sa  $n = 3$  odluče isto kao simulirani procesori, i njihove odluke će takođe biti korektne

# Algoritam eksponencijalnog stabla

11

- Ovaj algoritam koristi:
  - ▣  $f + 1$  rundi (optimalan)
  - ▣  $n = 3f + 1$  procesora (optimalan)
  - ▣ eksponencijalnu veličinu poruka (suboptimalan)
- Svaki procesor održava stablo u svom lokalnom stanju
- Vrednosti u stablu se popunjavaju tokom  $f + 1$  rundi
- Na kraju, vrednosti u stablu se koriste za odluku

# Lokalno stablo

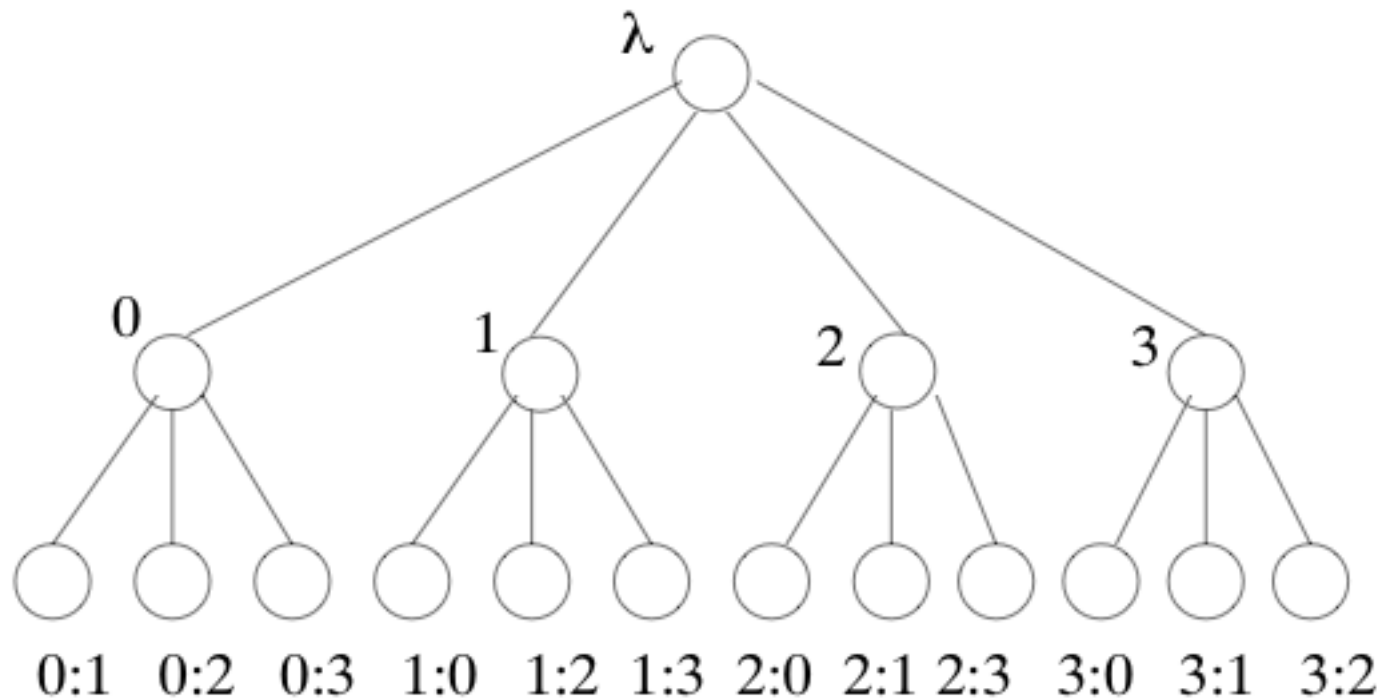
12

- Svaki čvor stabla je označen sa sekvencom jedinstvenih indeksa procesora
- Labela korena je prazna sekve.  $\lambda$ ; koren ima nivo 0
- Koren ima  $n$  potomaka, označenih sa 0 do  $n - 1$
- Čvor potomak označen sa  $i$  ima  $n - 1$  potomaka, označenih sa  $i : 0$  do  $i : n - 1$  (izostavlja se  $i : i$ )
- Čvor na nivou  $d$  označen sa  $v$  ima  $n - d$  potomaka, označenih sa  $v : 0$  do  $v : n - 1$  (izostavljajući bilo koji indeks koji se pojavljuje u  $v$ )
- Čvorovi na nivou  $f + 1$  su listovi

# Primer lokalnog stabla

13

Stablo za  $n = 4$  i  $f = 1$  :



# Popunjavanje čvorova stabla

14

- Inicijalno stavi svoj ulaz u koren (nivo 0)
- Runda 1:
  - ▣ šalji svima nivo 0 svog stabla
  - ▣ stavi vred  $x$  primljenu od svakog  $p_i$  u čvoru stabla označe. sa  $i$  (nivo 1); ako je potrebno, koristi podra. vred. (default)
  - ▣ " $p_i$  mi je rekao da je ulaz od  $p_i$  bio  $x$ "
- Runda 2:
  - ▣ šalji svima nivo 1 svog stabla
  - ▣ stavi vred  $x$  primljenu od svakog  $p_i$  za svaki čvor stabla  $k$  u čvoru označe. sa  $k : i$  (nivo 2); ili podrazumevanu vred.
  - ▣ " $p_i$  mi je rekao da je  $p_k$  rekao  $p_i$  da je ulaz od  $p_k$  bio  $x$ "
- Nastavi za  $f + 1$  rundi

# Odlučivanje

15

- U rundi  $f + 1$ , svaki procesor koristi vred u svom stablu da izračuna svoju odluku
- Rekurzivno računaj vred „rešenja“ za koren stabla,  $\text{resolve}(\lambda)$ , na osnovu vred „rešenja“ za druge čvorove stabla:

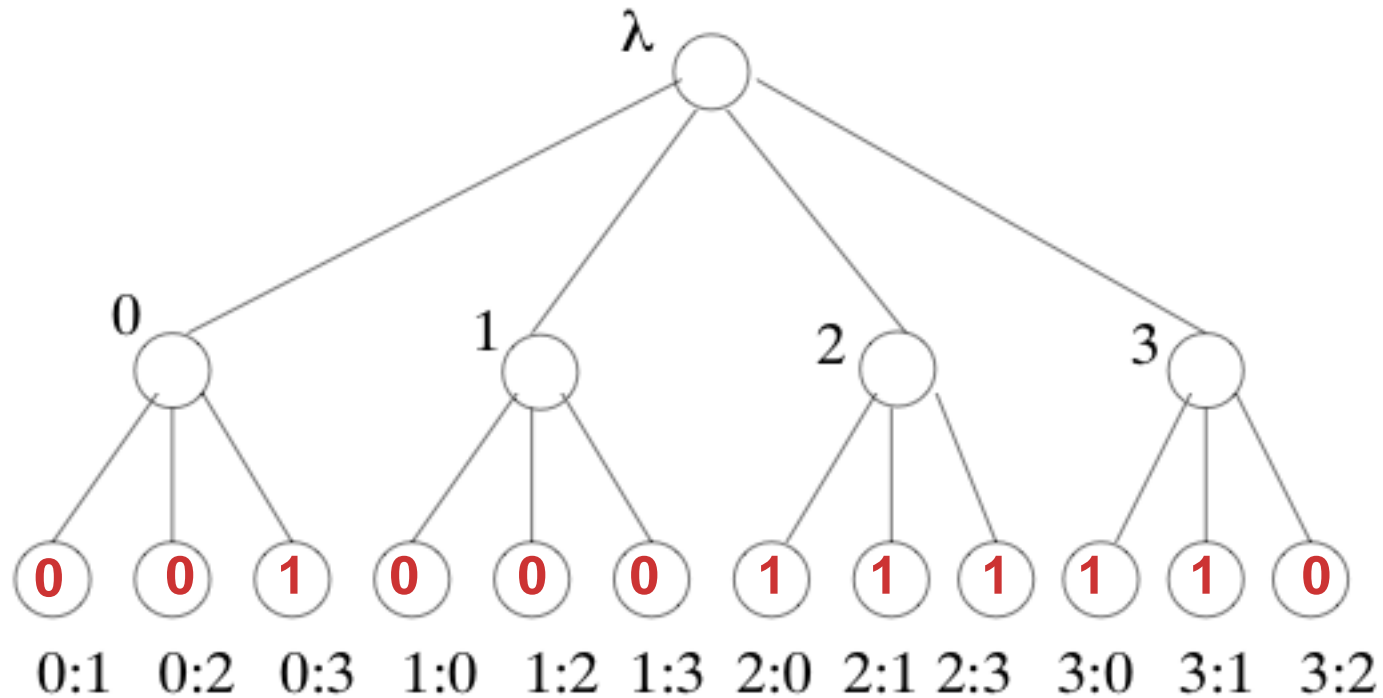
$$\text{resolve}(\pi) = \begin{cases} \text{vred u čvoru označe. sa } \pi \text{ ako je to list} \\ \text{majority}\{\text{resolve}(\pi') : \pi' \text{ je potomak od } \pi\} \text{ inače (koristi podraz. vr. ako nema većine)} \end{cases}$$



# Primer vrednosti rešenja

16

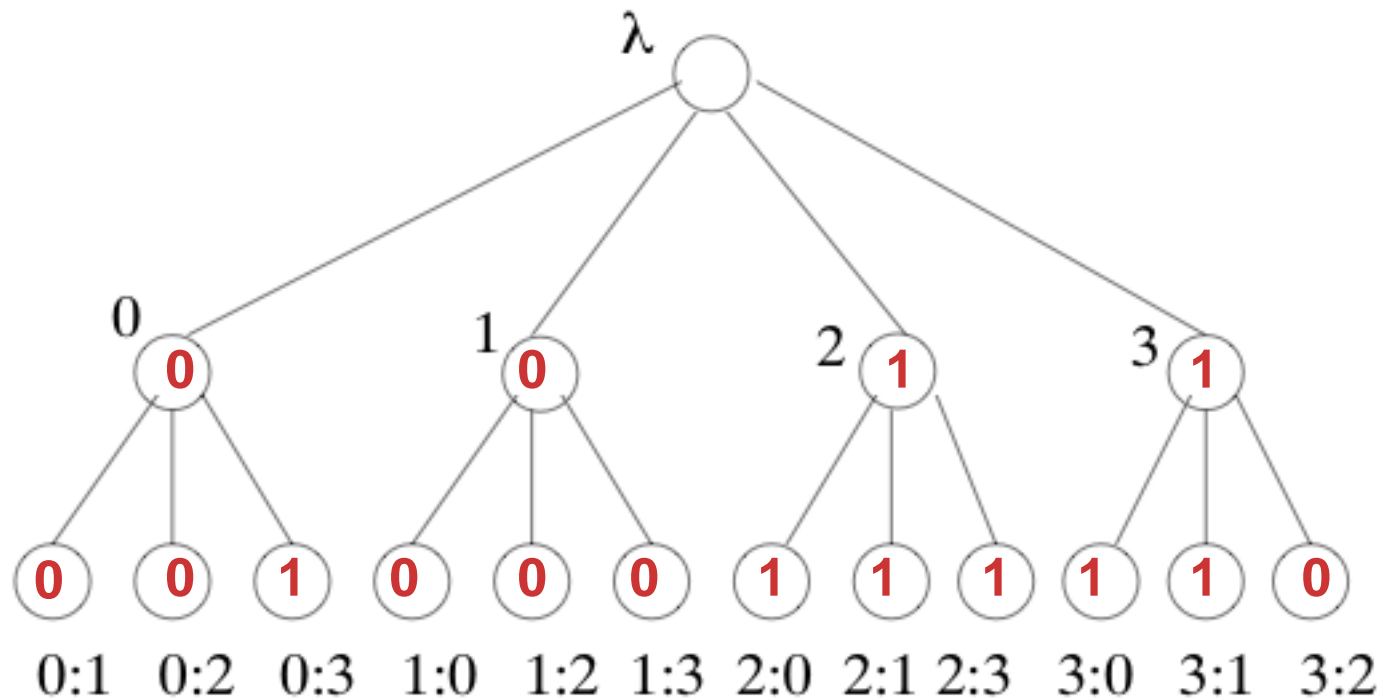
Stablo za  $n = 4$  i  $f = 1$  :



# Primer vrednosti rešenja

16

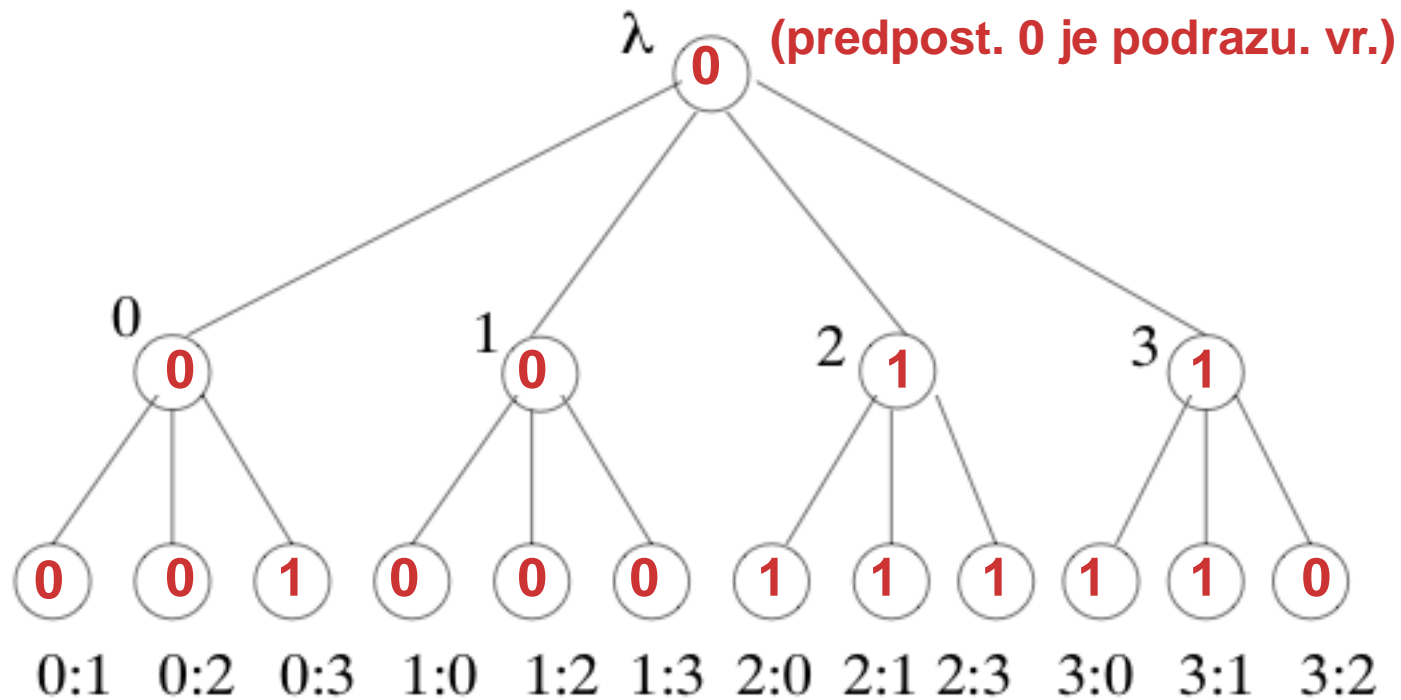
Stablo za  $n = 4$  i  $f = 1$  :



# Primer vrednosti rešenja

16

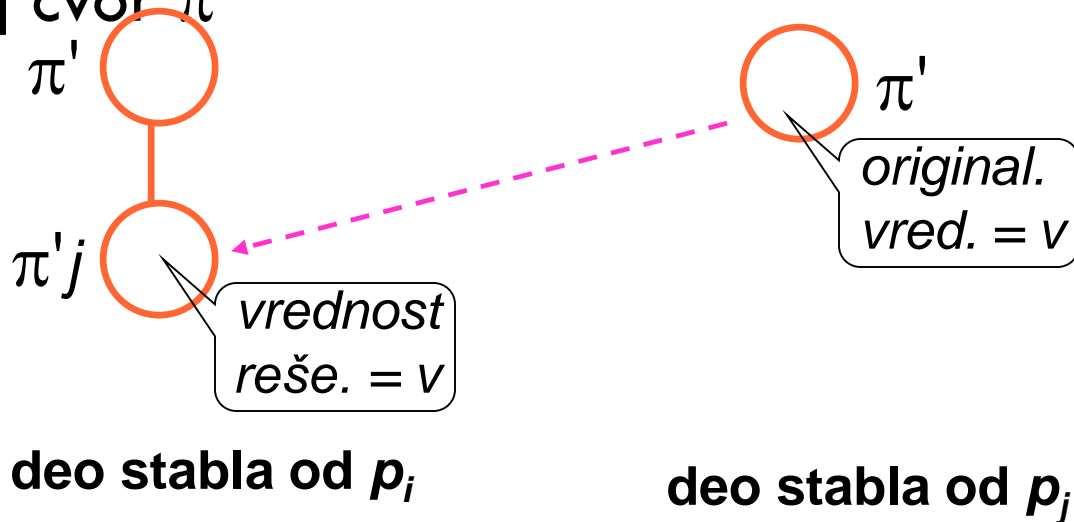
Stablo za  $n = 4$  i  $f = 1$  :



# Vrednosti rešenja su konzistentne

17

**Lema (10.9):** Ako su  $p_i$  i  $p_j$  ispravni, onda je  $p_i$ -jeva vred. rešenja za svoj čvor stabla označe.  $\pi' j$  (šta  $p_j$  kaže  $p_i$ -ju za čvor  $\pi'$ ) jednak onome što je  $p_j$  stavio u svoj čvor  $\pi'$



# Vrednosti rešenja su konzistentne

18

## Ideje za dokaz:

- Po indukciji na visinu čvora stabla
- Induktivna hipoteza da se zna da su vred. rešenja za potomke čvora, koji odgovaraju ispravnim proc, konzistentne
- Koristi činjenicu da  $n > 3f$  i činjenicu da svaki čvor ima bar  $n - f$  potomaka da bi se znalo da je većina potomaka ispravna

# Validnost

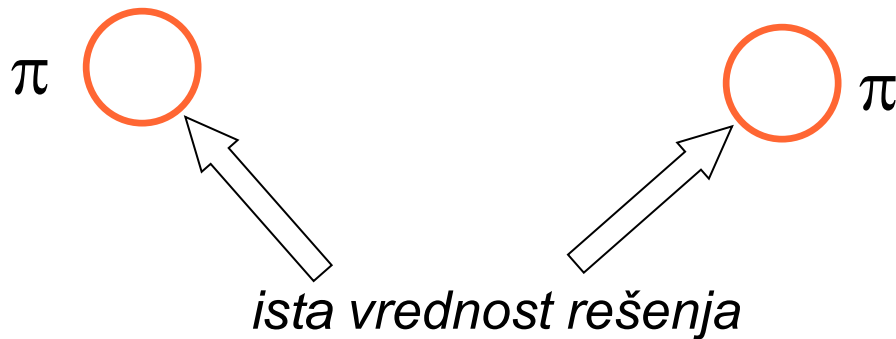
19

- Predpost. da su svi ulazi v
- Ispravan proc.  $p_i$  odlučuje  $\text{resolve}(\lambda)$ , što je većinska vred između  $\text{resolve}(j)$ ,  $0 \leq j \leq n-1$ , na osnovu stabla od  $p_i$
- Pošto su vrednosti rešenja konzistente,  $\text{resolve}(j)$  (u  $p_i$ ) je vred smeštena u koren stabla od  $p_j$ , što je ulazna vred od  $p_j$  ako je  $p_j$  ispravan
- Pošto postoji većina ispravnih procesora,  $p_i$  odlučuje v

# Zajednički čvorovi

20

- Čvor stabla  $\pi$  je **zajednički** ako svi ispravni proc. dobiju istu vrednost za  $\text{resolve}(\pi)$



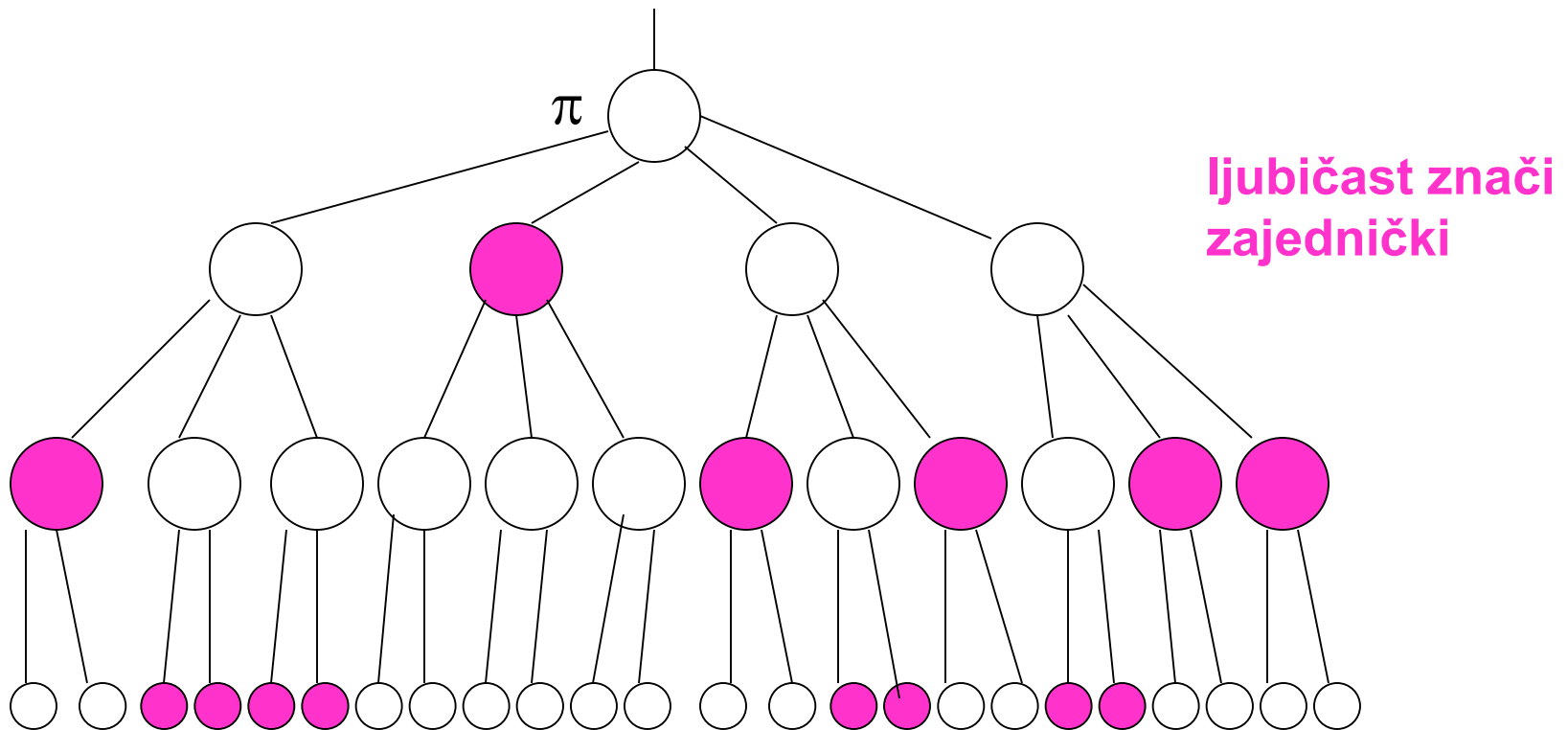
deo stabla od  $p_i$

deo stabla od  $p_j$

# Zajedničke granice

21

- Čvor stabla  $\pi$  ima **zajedničku granicu** ako svaka putanja iz  $\pi$  do lista sadrži neki zajednički čvor





# Zajednički čvorovi i granice

22

**Lema (10.10):** Ako  $\pi$  ima zajedničku granicu, onda je  $\pi$  zajednički

**Ideje za dokaz:** Po indukciji na visinu  $\pi$ . Koristiti činjenicu da je funkcija `resolve` definisana na osnovu većinske vrednosti

# Dogovor

23

- Čvorovi na svakoj putanji od potomka korena do lista potiču od  $f + 1$  različitih proc.
- Pošto postoji najviše  $f$  procesora u otkazu, bar 1 takav čvor odgovara ispravnom procesoru
- Ovaj čvor je zajednički (po lemi o konzistentnosti vrednosti rešenja)
- Zato koren ima zajedničku granicu
- Zato je koren zajednički (po predhodnoj lemi)

# Složenost

24

Algoritam eksponencijalnog stabla koristi

- $n > 3f$  procesora
- $f + 1$  rundi
- poruke eksponencijalne veličine:
  - ▣ svaka poruka u rundi  $r$  sadrži:  
 $n(n-1)(n-2)\dots(n-(r-2))$  vrednosti
  - ▣ Kad je  $r = f + 1$ , ta veličina je eksponencijalna ako je  $f$  više od konstante u odnosu na  $n$

# Jedan polinomijalan algoritam

25

- Veličina por se može redukovati na polinomijalnu pomoću jednog jednostavnog algoritma
- Broj procesora se povećava na
$$n > 4f$$
- Broj rundi se povećava na
$$2(f + 1)$$
- Koristi se  $f + 1$  faza, svaka uzima dve runde

# Algoritam Kralj (Faze)

26

**Kod za svaki procesor  $p_i$ :**

pref := moj ulaz

prva runda faze  $k$ ,  $1 \leq k \leq f+1$ :

šalji svima pref

primi pref vrednosti od drugih

neka je maj vred koja se pojavila  $> n/2$  puta (0 ako takve nema)

neka je mult broj pojava od maj

druga runda faze  $k$ :

**if**  $i = k$  **then** šalji svima maj // Ja sam kralj faze

primi tie-breaker (arbitražnu vred) od  $p_k$  (0 ako je nema)

**if** mult  $> n/2 + f$

**then** pref := maj

**else** pref := tie-breaker

**if**  $k = f + 1$  **then** odluči pref

# Lema anonimne faze

27

**Lema (10.12):** Ako svi ispravni procesori postavе  $\text{pref}$  na  $v$  na početku faze  $k$ , onda će svi to uraditi i na kraju faze  $k$

**Dokaz:**

- U prvoj rundi faze  $k$ , svaki ispravan proc. prima  $v$  bar  $(n - f)$  puta
- Pošto je  $n > 4f$ , sledi da je  $n - f > n/2 + f$
- Tako svaki ispravan proc. i dalje preferira  $v$

# Validnost kralja (faze)

28

Lema anonimne faze implicira validnost:

- Predpost. da svi proc. imaju ulaz  $v$
- Onda na početku faze 1, svi isp. proc. preferiraju  $v$
- Isto tako na kraju faze 1
- Isto tako na početku faze 2
- Isto tako na kraju faze 2
- ...
- Na kraju faze  $f + 1$ , svi isp. proc. preferiraju  $v$  i odlučuju  $v$

# Lema ispravnog kralja

29

**Lema (10.13):** Ako je kralj faze  $k$  ispravan, onda svi ispravni proc imaju istu preferencu na kraju faze  $k$

**Dokaz:** Neka su  $p_i$  i  $p_j$  ispravni

*Sluč. 1:*  $p_i$  i  $p_j$  koriste *tie-breaker* od  $p_k$ . Pošto je  $p_k$  ispravan, oni oba imaju istu preferencu



# Lema ispravnog kralja

30

*Sluč. 2:  $p_i$  koristi svoju većinsku vred  $v$  a  $p_j$  koristi kraljev *tie-breaker**

- Onda  $p_i$  prima više od  $n/2 + f$  preferenci za  $v$
- Zato  $p_k$  prima više od  $n/2 + f$  preferenci za  $v$
- Zato je *tie-breaker* od  $p_k$  jednak  $v$

# Lema ispravnog kralja

31

*Sluč. 3:*  $p_i$  i  $p_j$  koriste svako svoju većinsku vred.

- Predpost. da je većinska vred od  $p_i$  jednaka  $v$
- Onda  $p_i$  prima više od  $n/2 + f$  preferenci za  $v$
- Zato  $p_j$  prima više od  $n/2 + f$  preferenci za  $v$
- Zato je većinska vrednost od  $p_j$  takođe jednaka  $v$

# Dogovor

32

Pomoću predhodne 2 leme dokazujemo dogovor:

- Pošto postoji  $f + 1$  faza, bar jedna ima ispravnog kralja
- Lema ispravnog kralja implicira da na kraju te faze, svi ispravni procesori imaju istu preferencu
- Lema anonimne faze implicira da od te faze na dalje, svi ispravni procesori imaju istu preferencu
- Zati svi ispravni proc odlučuju isto

# Mere složenosti

33

- broj procesora  $n > 4f$
- $2(f + 1)$  rundi
- $O(n^2f)$  poruka, svaka veličine  $\log |V|$