

DISTRIBUIRANI ALGORITMI I SISTEMI

Otkazi procesora u slanju poruka

2

- **Ispad (Crash):** u nekoj tački procesor prestaje da izvodi korake
 - u poslednjem koraku, procesor može poslati neki podskup poruka koje bi trebao da pošalje
- **Vizantijski (Byzantine):** procesor proizvoljno menja stanje i šalje poruke sa proizvoljnim sadržajem

Problem konsenzusa

3

- Svaki procesor ima neku ulaznu vrednost
- *Završetak*: Konačno svaki ispravan procesor mora da se odluči za neku vrednost
 - ▣ odluka je neopoziva!
- *Dogovor*: Sve odluke od ispravnih procesora moraju biti iste
- *Validnost*: Ako su svi ulazi isti, onda odluka svakog ispravnog procesora mora biti jednaka zajedničkom ulazu

Primeri konsenzusa

4

- Binarni ulazi:
 - ulazni vektor 1,1,1,1,1
 - odluka mora biti 1
 - ulazni vektor 0,0,0,0,0
 - odluka mora biti 0
 - ulazni vektor 1,0,0,1,0
 - odluka mora biti ili 0 ili 1
- Ulazi sa više od 2 vrednosti:
 - ulazni vektor 1,2,3,2,1
 - odluka može biti 1 ili 2 ili 3

Pregled postojećih rezultata

5

- Sinhroni sistemi
- Najviše f procesora u otkazu
- Uske granice za slanje poruka:

	Otkazi ispada	Vizantijski otkazi
broj rundi	$f + 1$	$f + 1$
ukupan broj procesora	$f + 1$	$3f + 1$
veličina poruke	polinomijalna	polinomijalna

Pregled postojećih rezultata

6

- *Nije moguć* u asinhronom slučaju
- Čak i ako hoćemo da tolerišemo samo jedan otkaz tipa ispada
- Važi i za sisteme sa slanjem poruka i za deljenu memoriju

Modeliranje otkaza tipa ispada

7

- Menjamo definicije **prihvatljivog** izvršenja da bi obuhvatili otkaze tipa ispada:
- Svi osim skupa od najviše f procesora (oni u **otkazu**) izvode beskonačan broj koraka
 - U *sinhronom* slučaju: nakon što procesor ne izvede neki korak, on više ne izvodi ni jedan korak
- U poslednjem koraku, **proizvoljan podskup** odlaznih poruka uspeva da uđe u kanale

Modeliranje vizantijskih otkaza

8

- Menjamo definicije **prihvatljivog** izvršenja da bi obuhvatili vizantijske otkaze:
- Skup od najviše f procesora (onih u **otkazu**) može slati poruke proizvoljnog sadržaja i menjati stanje proizvoljno (tj., ne po svojim prelaznim funkcijama)

Algoritam konsenzusa za ispade

9

Kod za svaki procesor:

$v :=$ moj ulaz

u svakoj rundi od 1 do $f+1$:

ako još nije poslao v , onda šalji v svima

čekaj prijem poruka za ovu rundu

$v :=$ minimum od svih primljenih vrednosti i
tekuće vrednosti v

ako je ovo runda $f+1$, onda odluči v

Izvršenje algoritma

10

- runda 1:
 - šalji svoj ulaz
 - primi poruke runde 1
 - računaj vrednost za v
 - runda 2:
 - šalji v (ako je ovo nova vred)
 - primi poruke runde 2
 - računaj vrednost za v
 - ...
 - runda $f + 1$:
 - šalji v (ako je ovo nova vred)
 - primi poruke runde $f + 1$
 - računaj vrednost za v
 - odluči v
- Relacija sa formalnim modelom
- inicijalno u kanalima
dogadjaji isporuke
dogadjaji računanja
- zbog predhodnih dogadjaja računanja
dogadjaji isporuke
dogadjaji računanja
- zbog predhodnih dogadjaja računanja
dogadjaji isporuke
dogadjaji računanja
deo dogadjaja računanja

Korektnost algoritma konsenzusa za otkaze tipa ispada

11

Završetak: Po kodu, završava se u rundi $f+1$

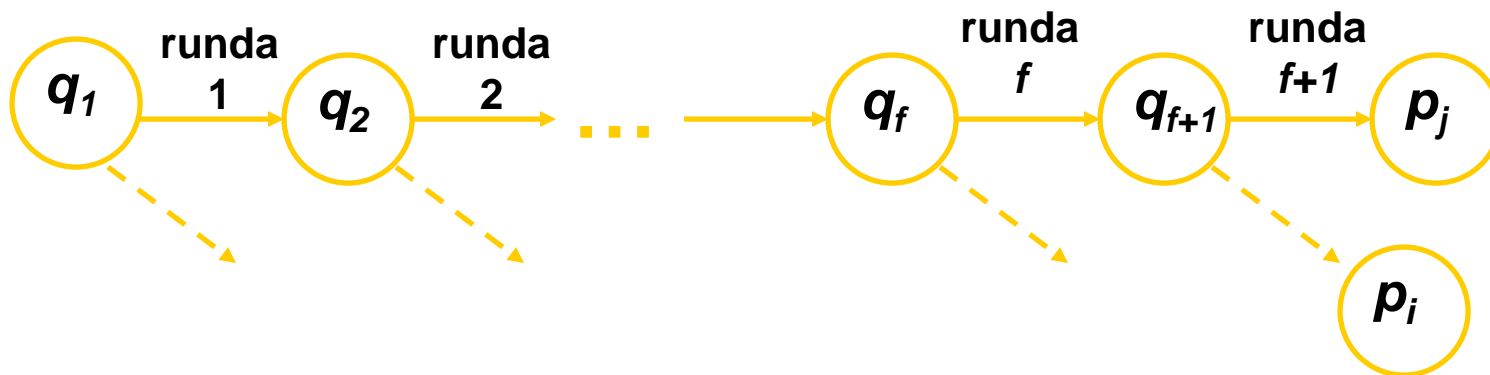
Validnost: Važi jer procesori ne unose lažne poruke:
ako su svi ulazi isti, onda je to jedina vrednost u
optičaju

Korektnost algoritma konsenzusa za otkaze tipa ispada

12

Dogovor:

- Predpost. radi kontradikcije da se p_j odluči za manju vred, x , nego p_i
- Onda je x bio sakriven od p_i sa lancem procesora u otkazu:



- Postoji $f + 1$ neispravnih procesora u ovom lancu, kontradikcija

Performansa algoritma konsenzusa za otkaze tipa ispada

13

- Broj procesora $n > f$
- $f + 1$ rundi
- najviše $n^2 \cdot |V|$ poruka, svaka veličina $\log |V|$ bita, gde je V ulazni skup

Donja granica za runde

14

Predpostavke:

- $n > f + 1$
- svaki procesor treba da pošalje poruku svakom drugom procesoru u svakoj rundi
- ulazni skup je $\{0,1\}$

Izvršenja sa retkim otkazima

15

- Loše ponašanje za predhodni algoritam je bilo kad je bio jedan ispad po rundi
- Ovo je loše generalno
- **Izvršenja sa retkim otkazima** imaju najviše jedan ispad po rundi
- U ovo dokazu bavićemo se isključivo sa izvršenjima sa retkim otkazima

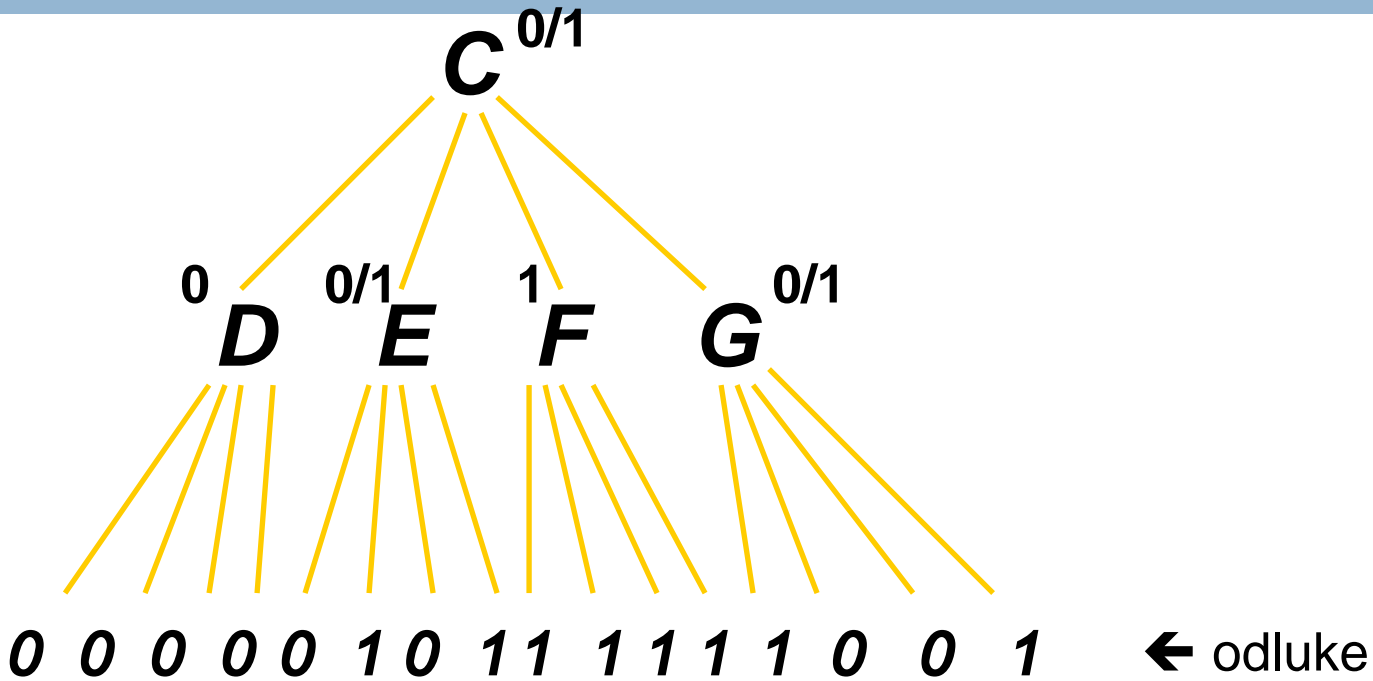
Valenca konfiguracije

16

- **Valenca** konfiguracije C je skup svih vrednosti za koje su se odlučili ispravni procesori u konfiguracijama dostupnim iz C putem prihvatljivog izvršenja (sa retkim otkazima)
- **Bivalentan**: skup sadrži 0 i 1
- **Univalentan**: skup sadrži samo jednu vrednost
 - 0-valentan ili 1-valentan

Valenca konfiguracije

17



0/1 : bivalentan

1 : 1-valentan

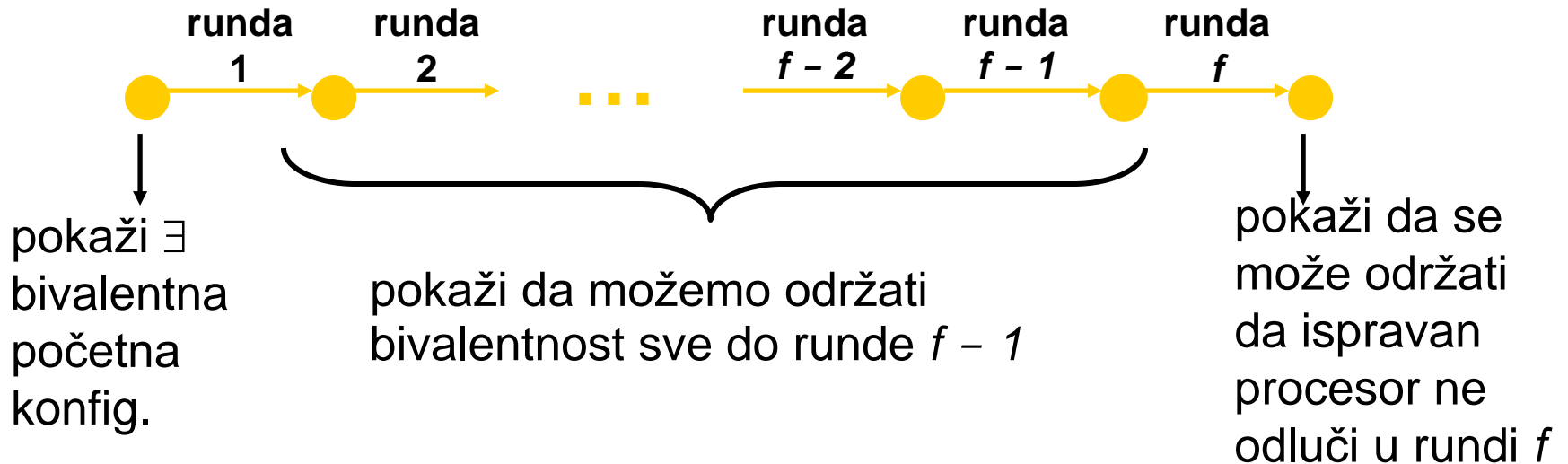
0 : 0-valentan

Iskaz o donjoj granici za runde

18

Teorema (10.3): Bilo koji f -elastičan algoritam konsenzusa zahteva bar $f + 1$ rundi u najgorem slučaju

Strategija dokaza:



Postojanje bivalentne početne konfiguracije

19

- Predpostavimo radi kontradikcije da su sve početne konfiguracije univalentne

ulazi	valence
000...00	0
000...01	?
000...11	?
...	
001...11	?
011...11	?
111...11	1

po uslovu validnosti



Postojanje bivalentne početne konfiguracije

20

- Predpostavimo radi kontradikcije da su sve početne konfiguracije univalentne

inputs	valency
000...00	0
000...01	?
000...11	?
...	
001...11	? 0
011...11	? 1
111...11	1

po uslovu validnosti

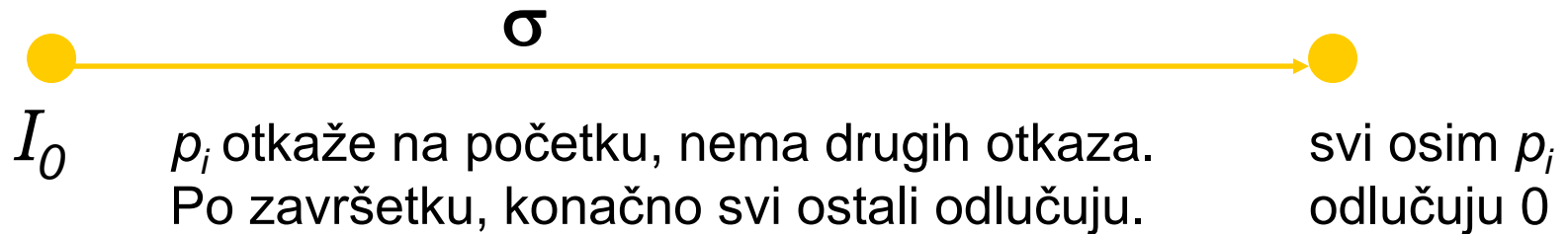
Postoje 2 susedne konfiguracije sa razlicitim valencama

Postojanje bivalentne početne konfiguracije

21

□ Neka je:

- I_0 neka 0-valentna početna konfiguracija
- I_1 neka 1-valentna početna konfiguracija
- takve da se razlikuju samo u ulazu od p_i

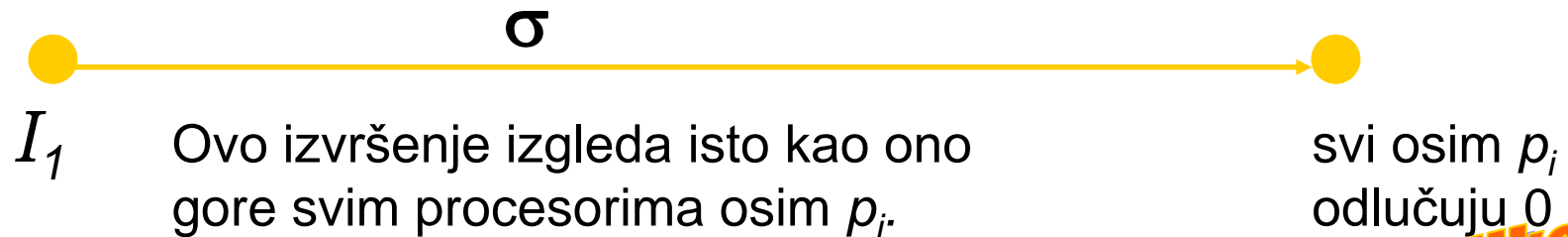
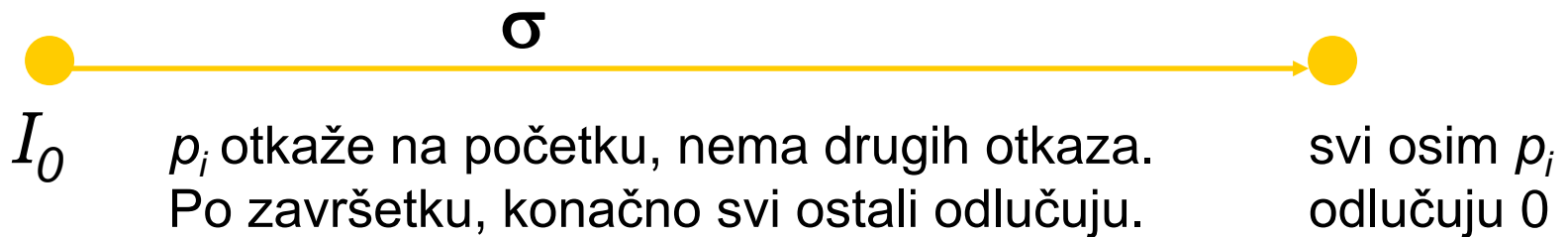


Postojanje bivalentne početne konfiguracije

22

□ Neka je

- I_0 je neka 0-valentna početna konfiguracija
- I_1 je neka 1-valentna početna konfiguracija
- takve da se razlikuju samo u ulazu od p_i



Kontradikcija!