

DISTRIBIURANI ALGORITMI I SISTEMI

Distribuirani Sistemi (DS)

2

- Distribuirani sistemi su ušli u široku upotrebu:
 - deljenje resursa
 - komunikacija
 - poboljšana performanse
 - brzina
 - otpornost na otkaze (fault tolerance)
- Karakterišu ih:
 - nezavisne aktivnosti (konkurencija)
 - slabo spregnut paralelizam (heterogenost)
 - inherentna neodređenost

Neodređenost u DS

3

- Neodređenost potiče od:
 - različitih brzina procesora
 - promenljivog komunikacionog kašnjenja
 - (delimičnih) otkaza
 - višestrukih ulaznih tokova i interaktivnih ponašanja

Rasudivanje o DS

4

- Neodređenost otežava proveru korektnosti sistema
- Savladavanje ovih teškoća:
 - ▣ identifikacija i apstrakcija osnovnih problema
 - ▣ precizna formulacija problema
 - ▣ projektovanje algoritama za rešavanje problema
 - ▣ dokazivanje korektnosti algoritama
 - ▣ analiza složenosti algoritama (npr., vreme, prostor, poruke)
 - ▣ dokazivanje nemogućih rezultata i donjih granica

Moguća korist od teorije

5

- pažljive specifikacije razjašnjavaju nameru
- povećana sigurnost u korektnost
- ako je apstrakcija dobra, rezultati su relevantni u mnogim situacijama
- Saznanja o inherentnim ograničenjima
 - ▣ npr. NP-potpunost

Oblasti primene

6

- Oblasti iz kojih potiču klasični problemi distribuiranog/konkurentnog računanja:
 - ▣ operativni sistemi
 - ▣ (distribuirane) baze podataka
 - ▣ softver otporan na otkaze
 - ▣ komunikacione mreže
 - ▣ multiprocesorske arhitekture
- Novije oblasti primene:
 - ▣ računanja u klauđu (cloud computing)
 - ▣ mobilna računanja, ...

Osnovni modeli DS (1 / 2)

7

- Uvode se dva osnovna komunikaciona modela:
 - sa slanjem poruka
 - sa deljenom memorijom
- i dva osnovna vremenska modela:
 - sinhroni
 - asinhroni

Osnovni modeli DS (2/2)

8

Slanje poruka Deljena memorija

sinhroni	Da	Ne
asinhroni	Da	Da

(Sinhroni model sa deljenom memorijom je PRAM)

Pregled tema: Deo I (osnove)

9

- Ovde spadaju kanonički problemi i pitanja:
 - graf algoritmi
 - izbor lidera
 - međusobno isključivanje
 - konsenzus otporan na otkaze
 - uzročnost (causality) i vreme

Pregled tema: Deo II (Simulacije)

10

- „Simulacije“ su apstrakcije, ili tehnike, koje dati model pretvore u jednostavniji model. Na primer:
 - slanje svima i slanje u grupi
 - distribuirana deljena memorija
 - jače vrste deljenih promenljivih
 - više sinhronizma
 - više benignih otkaza

Pregled tema: Deo III (Napredne teme)

11

- **Nastavci u nekim već uvedenim pravcima:**
 - nasumični algoritmi (randomized algorithms)
 - jače vrste deljenih objekata proizvoljne vrste
 - koje vrste problema su rešive u asinhronim sistemima
 - detektori otkaza
 - samo-stabilizacija

Odnos teorije i prakse

12

- OS sa deljenjem vremena: pitanja u vezi (virtuelne) konkurencije procesa kao što su:
 - ▣ međusobno isključivanje
 - ▣ međusobno blokiranje (deadlock)takođe se pojavljuju u DS
- MIMD multiprocesori:
 - ▣ nema zajedničkog takta => asinhroni model
 - ▣ zajednički takt => sinhroni model
- slabo spregnute mreže, ala Internet, => asinhroni model

Odnos teorije i prakse

13

- Modeli otkaza:
 - ispad: procesor u otkazu jednostavno stane. Idealizacija stvarnosti.
 - Vizantijski (proizvoljno): konzervativna pretpostavka, kada je model otkaza nepoznat ili zlonameran.
 - Samo-stabilizacija: algoritam se automatski oporavlja iz prolaznog stanja korupcije; potreban je za aplikacije koje se dugo izvršavaju.

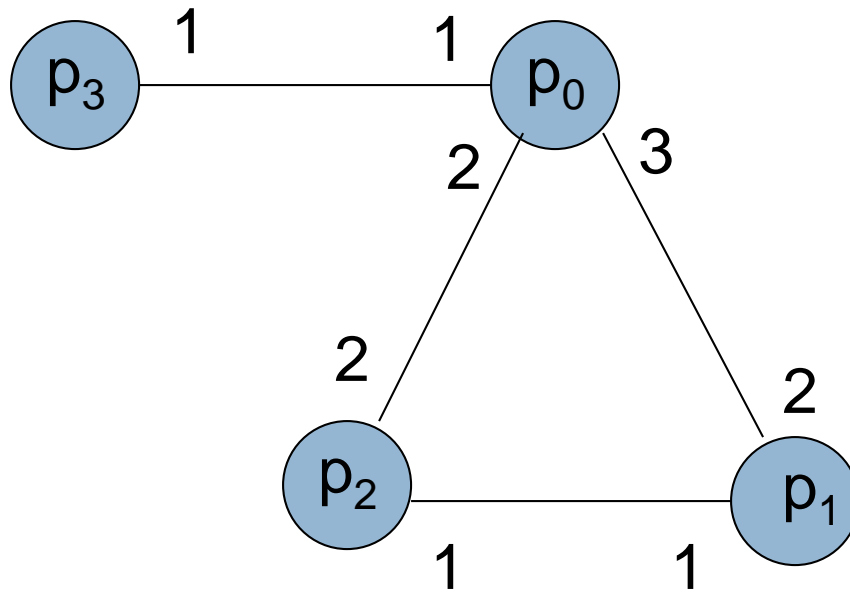
Model sa slanjem poruka

14

- procesori su p_0, p_1, \dots, p_{n-1} (čvorovi grafa)
- dvosmerni kanali od-tačke-do-tačke (neusmereni lukovi grafa)
- svaki procesor označava svoje strane kanala 1, 2, 3,...; može da nezna ko je na drugom kraju

Model sa slanjem poruka

15



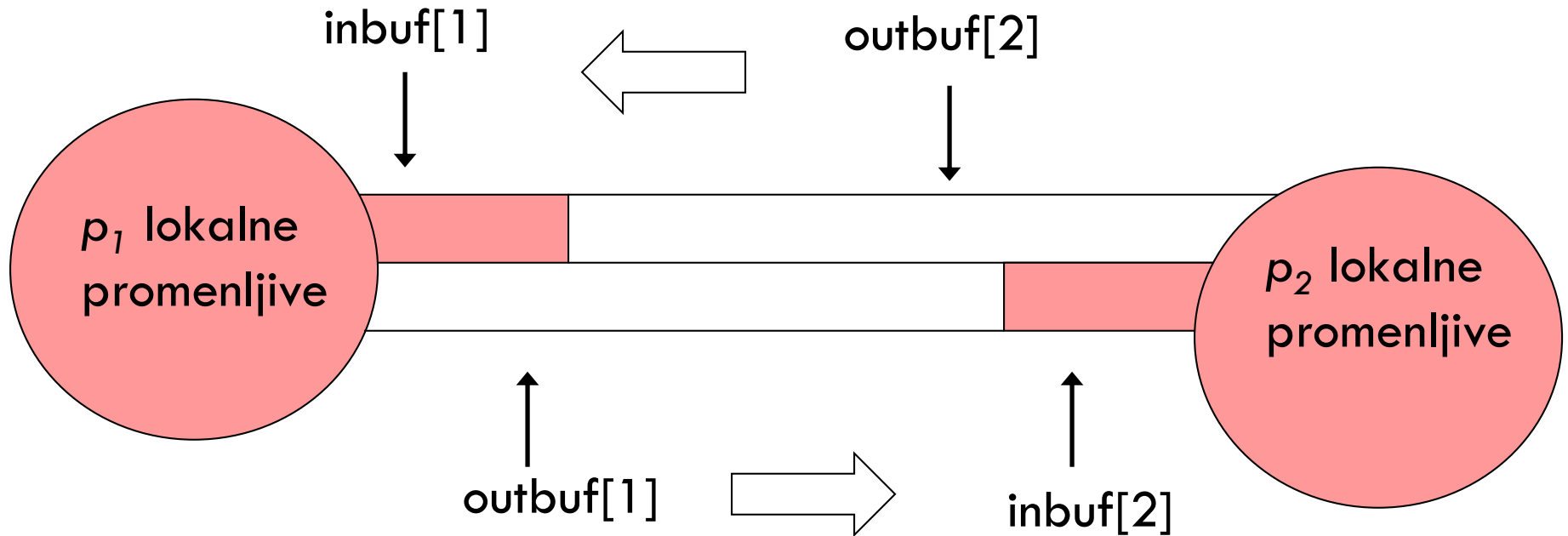
Modeliranje procesora i kanala

16

- Procesor je automat (state machine) koji čine
 - ▣ lokalna stanja procesora
 - ▣ mehanizmi za modeliranje kanala
- Kanal usmeren od procesora p_i ka procesoru p_j modelira se sa dve promenljive:
 - ▣ *outbuf* promenljiva od p_i i
 - ▣ *inbuf* promenljiva od p_j
- *Outbuf* odgovara fizičkom kanalu, a *inbuf* odgovara redu dolaznih poruka.

Modeliranje procesora i kanala

17



Obojena oblast (lokalne prom + inbuf) je *dostupno* stanje procesora.

Konfiguracija

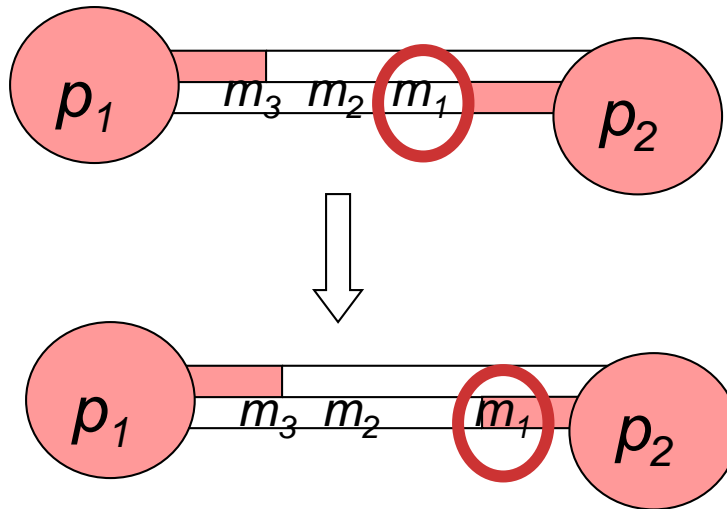
18

- Vektor stanja procesora (uključujući outbufs, tj. kanale), je *konfiguracija* sistema (elementi su stanja pojedinačnih procesora).
- Tekući snimak (snapshot) sistema sadrži: dostupna stanja procesora (lokalne prom + redovi dolaznih poruka) kao i komunikacioni kanali.

Događaj isporuke (poruke)

19

- Prebacuje poruku iz outbuf pošiljaoca u inbuf primaoca; poruka će biti raspoloživa u sledećem koraku primaoca.



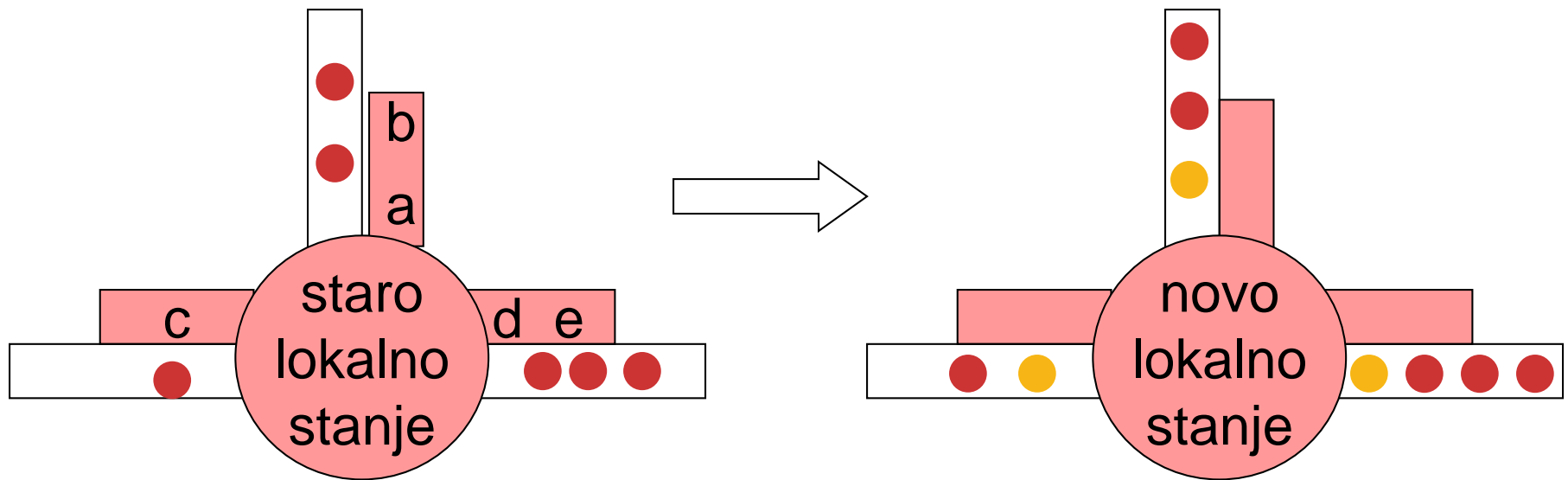
Događaj računanja

20

- Dešava se u jednom procesoru.
- Počinje sa starim dostupnim stanjem (lokalne prom + dolazne poruke).
- Primeni se funkcija prelaza iz automata za dati procesor; obrade se sve dolazne poruke.
- Završava sa novim dostupnim stanjem, sa praznim inbufs, i novim odlaznim porukama.

Događaj računanja

21



ljubičasto označava dostupno stanje: lokalne prom i dolazne poruke
belo označava bafere odlaznih poruka

Izvršenje

22

- Format je
 - config, event, config, event, config, ...
- u prvom config: svaki procesor je u početnom stanju i svi inbufs su prazni
- za svaki susedan (config, event, config), novi config je isti kao stari config osim ako je *event*:
 - ▣ događaj isporuke: zadana poruka se prenosi iz outbuf pošiljaoca u inbuf primaoca
 - ▣ događaj računanja: zadato stanje procesora (uključujući outbufs) menja se prema funkciji prelaza

Prihvatljivost

23

- Def. izvršenja daje neke osnovne „sintaksne“ uslove
 - ▣ obično uslove *sigurnosti* (true u svakom konačnom prefiksu)
- Ponekad hoćemo da postavimo dodatna ograničenja
 - ▣ obično uslovi *životnosti* (na kraju se nešto desi)
- Izvršenja koja zadovolje dodatna ograničenja su *prihvatljiva*; to su izvršenja koja moraju da reše problem koji je od interesa.
 - ▣ Definicija značenja “prihvatljiv” se menja od konteksta do konteksta, zavisno od detalja onoga šta se modelira.

Asinhrona izvršenja

24

- Izvršenje je *prihvatljivo za asinhroni model* ako:
 - svaka poruka iz outbuf na kraju bude isporučena
 - svaki procesor obavlja neograničen broj koraka.
- Nema ograničenja na to kada se događaji dešavaju: nisu isključena proizvoljna kašnjenja poruka i relativne brzine procesora.
- Modelira se pouzdan sistem (nema gubitaka poruka i ni jedan procesor ne prestaje da radi).

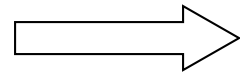
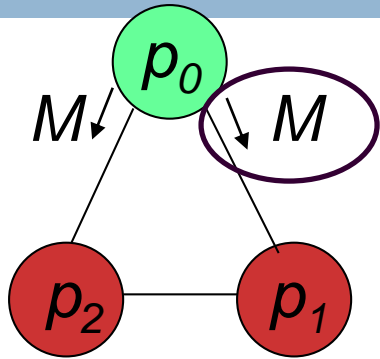
Primer: Plavljenje (Flooding)

25

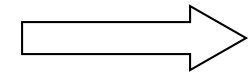
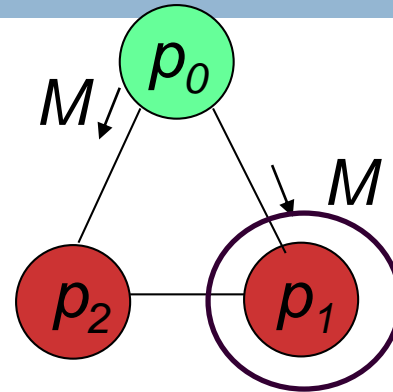
- Opiši jednostavn algoritma plavljenja kao kolekciju automata u interakciji.
- Lokalno stanje svakog procesora sadrži prom *color*, koja ima vrednost bilo **crveno** ili **zeleno**.
- Na početku:
 - ▣ p_0 : *boja* = **zeleno**, svi outbufs sadrže M
 - ▣ drugi: *boja* = **crveno**, svi outbufs prazni
- Prelaz: Ako je M u inbuf i *boja* = **crveno**, onda promeni *boju* u **zeleno** i pošalji M u sve outbufs.

Primer: Plavljenje (1/2)

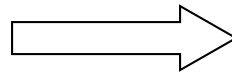
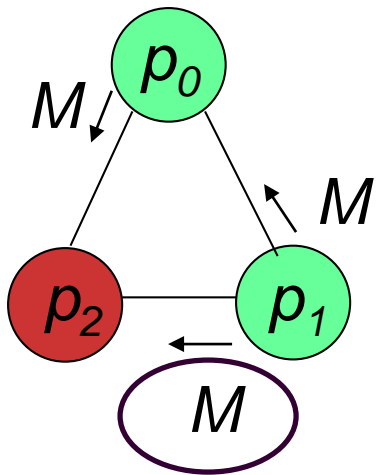
26



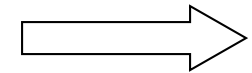
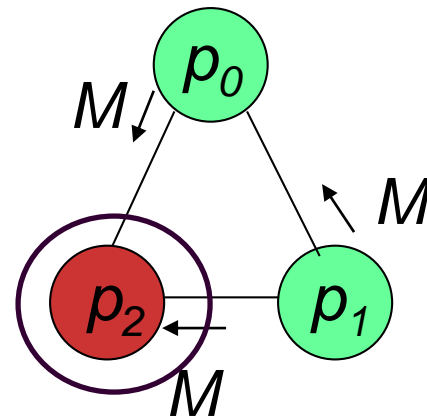
Isporuka M
iz p_0 na p_1



Računanje
na p_1



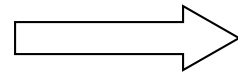
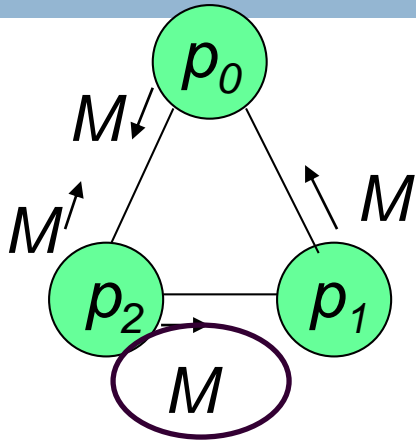
Isporuka M
iz p_1 na p_2



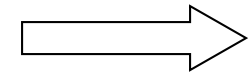
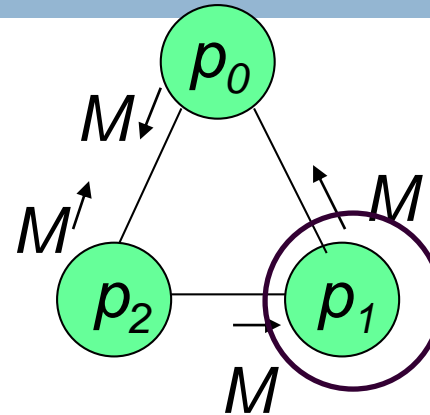
Računanje
na p_2

Primer: Plavljenje (2/2)

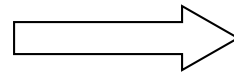
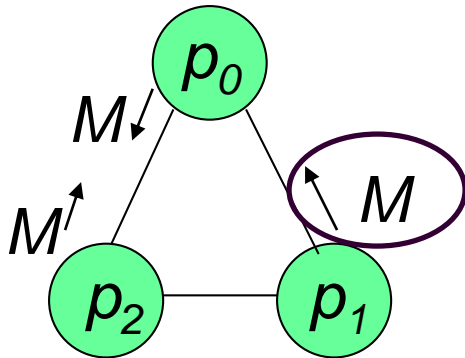
27



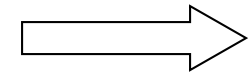
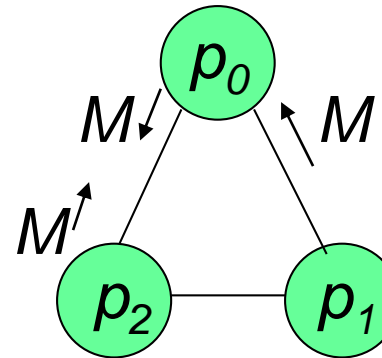
Isporuka M
iz p_2 na p_1



Računanje
na p_1



Isporuka M
iz p_1 na p_0



itd. do
isporuke
ostalih
poruka

Nedeterminizam

28

- Predhodno izvršenje nije jedino prihvatljivo izvršenje algoritma Plavljenje na tom trouglu.
- Ima ih nekoliko, zavisno od redosleda u kom se poruke isporučuju.
- Npr., poruka od p_0 je mogla stići do p_2 pre poruke od p_1 .

Završetak (Termination)

29

- Iz tehničkih razloga, prihvatljiva izvršenja se definišu kao beskonačna.
- Međutim, algoritmi se često završavaju.
- Radi modeliranja završetka algoritma, identifikuju se stanja *završetka* rada procesora: stanja u koja kad se uđe više se ne može izaći.
- Izvršenje je *završeno* kada je rad svih procesora završen i nema poruka u tranzitu (u inbufs ili outbufs).

Završetak algoritma Plavljenja

30

- Definišimo stanja *završetka* rada procesora kao ona u kojima je *boja* = zelena.

Metrika složenosti komunikacije

31

- **Složenost komunikacije:** max broj poruka poslatih u bilo kom prihvatljivom izvršenju.
- To je metrika najgoreg slučaja.
- Kasnije će biti reči o metrikama prosečnog slučaja.

Složenost komunikacije za algoritam Plavljenje

32

- Složenost komunikacije: jedna poruka se šalje preko svakog luka u svakom smeru. Br. poruka je $2m$, gde je $m =$ broj lukova grafa.

Metrika složenosti obrade

33

- Kako meriti vreme u asinhronim izvršenjima?
- Napraviti **vremensko izvršenje** dodelom neopadajućih realnih vremena događajima, tako da je vreme između slanja i prijema bilo koje poruke *najviše 1*.
- Time se normalizuje najveće kašnjenje poruke unutar izvršenja da bude jedna jedinica vremena; ali i dalje se dopušta proizvoljno učešljavanje događaja.
- **Složenost obrade:** max vreme do završetka u bilo kom *vremenski* prihvatljivom izvršenju.

Složenost obrade za algoritam Plavljenje

34

- Setimo se da su stanja završetka rada procesora ona stanja u kojima je *boja* = zeleno.
- Složenost obrade: *dijametar* + 1 vrem. jedinca. (Čvor postaje zelen kada ga dostigne „lanac“ poruka iz p_0 .)
 - ▣ *Dijametar* grafa je maksimum najkraćeg puta od čvora v do čvora w , preko svih čvorova v i w unutar grafa.

Sinhroni sistemi sa slanjem poruka (1 / 2)

35

- Izvršenje je **prihvatljivo** za sinhroni model ako je ono beskonačna sekvenca „rundi“.
- Šta je „runda“?
- To je sekvenca događaja isporuke koji prebacuju sve poruke u tranzitu u inbuf, praćena sekvencom događaja računanja, po jedan za svaki procesor.

Sinhroni sistemi sa slanjem poruka (2/2)

36

- Nova def. prihvatljivosti obuhvata osobinu istovremenog marša poruka kod sinhronog modela.
- Ova def. takođe implicira sledeće:
 - ▣ svaka poslata poruka je isporučena
 - ▣ svaki procesor izvodi beskonačan broj koraka.
- **Vreme** se meri kao broj rundi do završetka.

Primer sinhronog modela (1 / 3)

37

- Neka se alogoritam plavljenja izvršava u sinhronom modelu na trouglu.
- Runda 1:
 - isporuka M od p_0 na p_1
 - isporuka M od p_0 na p_2
 - p_0 ne radi ništa (jer nema dolaznih poruka)
 - p_1 prima M , postaje **zelen** i šalje M ka p_0 i p_2
 - p_2 prima M , postaje **zelen** i šalje M ka p_0 i p_1

Primer sinhronog modela (2/3)

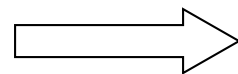
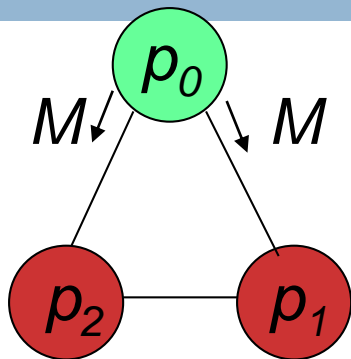
38

□ Runda 2:

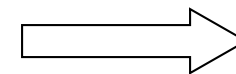
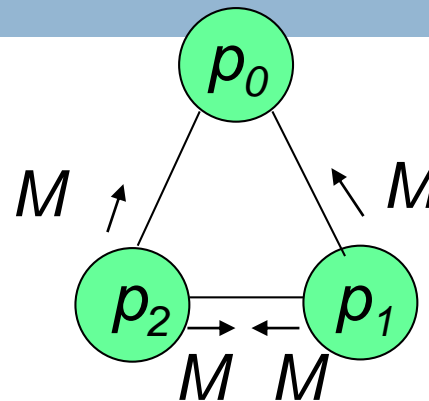
- isporuka M od p_1 na p_0
- isporuka M od p_2 na p_0
- isporuka M od p_2 na p_1
- isporuka M od p_1 na p_2
- p_0 ne radi ništa jer mu je vred. prom. *boja* već **zelen**
- p_1 ne radi ništa jer mu je vred. prom. *boja* već **zelen**
- p_2 ne radi ništa jer mu je vred. prom. *boja* već **zelen**

Primer sinhronog modela (3/3)

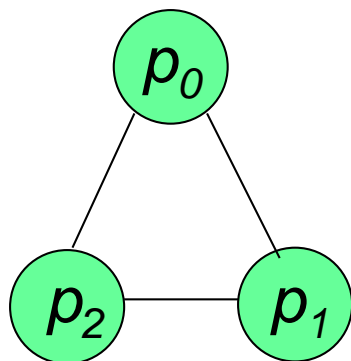
39



dogadaji
runde 1



dogadaji
runde 2



Složenost za sinhroni algoritam Plavljenje

40

- Posmatraju se samo izvršenja koja su prihvatljiva za sinhroni model (tj. koja zadovoljavaju def. sinhronog modela)
- Složenost obrade (vreme) je *dijametar + 1*
- Složenost komunikacije (br. poruka) je $2m$
- Isto kao u asinhronom slučaju.
- Što nije uvek slučaj za sve algoritme.