



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Konsensusongelma hajautetuissa järjestelmissä

Niko Välimäki

30.11.2007

Hajautetut algoritmit -seminaari





Konsensusongelma

- Päätöksen muodostaminen hajautetussa järjestelmässä
 - Prosessien välinen viestintä on *epäluotettava*
- Esimerkiksi: koordinoitun hyökkäyksen järjestäminen
- Kenraalit yrittävät nujertaa yhteisen vihollisen
 - Taistelu on voitettavissa vain, jos kaikki hyökkäävät
- Jokaisen kenraalin tulisi päättää: hyökätä vai ei?
 - Kenraali voi hyökätä vain, jos oma armeija on valmiina
 - Aluksi tiedetään vain oman armeijan tilanne
- Kenraalien välinen yhteydenpito on *epäluotettava*



Konsensusongelma: tietojenkäsittelytieteessä

- Hajautettu tietokantajärjestelmä
- Kun tietokantaan tehdään muutoksia, prosessien tulee joko hyväksyä tai hylätä muutokset
- Muutokset tulisi hyväksyä aina, kun mahdollista
 - ...ja vastaavasti hylätä aina, jos jokin prosesseista ei hyväksy muutoksia
- Tietokantojen sisältö pysyy yhtenäisenä vain, jos lopputulos on yksimielinen



Konsensusongelma: hajautettu järjestelmä

- *Täydellinen*, suuntaamaton verkko, jossa n prosessia
 - Toimii myös yleisemmässä tapauksessa
- Jokaisella prosessilla alkutila $alku_i \in \{0, 1\}$
 - 1 hyväksyvä tila, 0 hylkäävä tila
- Suoritus etenee synkronisesti: jokaisella *kierroksella* suoritetaan seuraavat kaksi askelta
 1. Jokainen prosessi lähettää viestinsä
 2. Prosessit vastaanottavat saapuvat viestit ja muuttavat tarvittaessa nykyistä tilaansa
- Prosessien väliset viestit voivat kadota



Konsensusongelma: esitelmän rakenne

- Deterministinen algoritmi
 - Ratkeamattomuustodistuksen idea
- Satunnaistettu algoritmi
 - Lähes optimaalinen ratkaisu
 - Teoreettinen alaraja



Deterministinen algoritmi: ehdot

- Määritellään seuraavat ehdot, joiden tulee täytyä:
 - (a) Prosessit tekevät lopullisen päätöksen, $tulos_i \in \{0, 1\}$, äärellisen monen kierroksen kuluessa
 - (b) Kaikki prosessit päätyvät samaan lopputulokseen $tulos_i$
 - (c) Kaikkien alkutilojen ollessa 0 myös kaikki $tulos_i = 0$
 - (d) Kaikkien alkutilojen ollessa 1 myös kaikki $tulos_i = 1$, jos kaikki viestit on toimitettu onnistuneesti
- Triviaalit ratkaisut, joissa $tulos_i$ on aina joko 1 tai 0, poissuljetaan ehdoilla (c) ja (d)

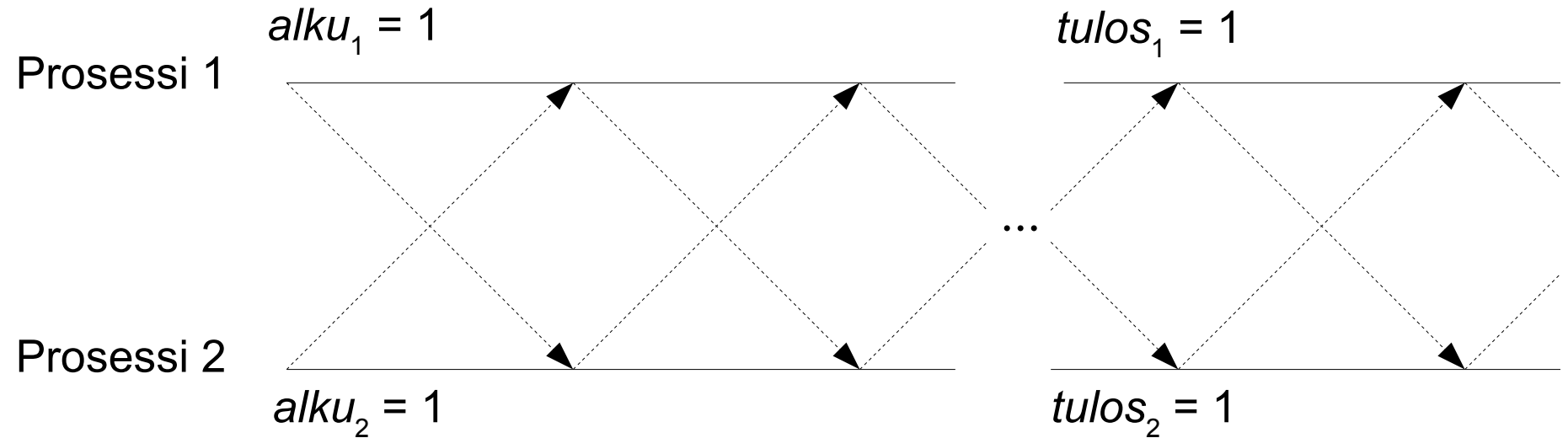


Ratkeamattomuustodistus: vastaoletus

- “On olemassa algoritmi A , joka ratkaisee konsensusongelman”
- Tarkastellaan kahden prosessin verkkoa
 - Prosessien alkutilat: $alku_1 = alku_2 = 1$
- *Suoritus* on jono järjestelmän tiloja (algoritmillä A)
 - Riippuu alkutiloista ja järjestelmässä toimitetuista viesteistä
- Olkoon suoritus α sellainen, että kaikki prosessien väliset viestit toimitetaan onnistuneesti
 - Vastaoletuksen ja ehdon (d) perusteella $tulos_1 = tulos_2 = 1$



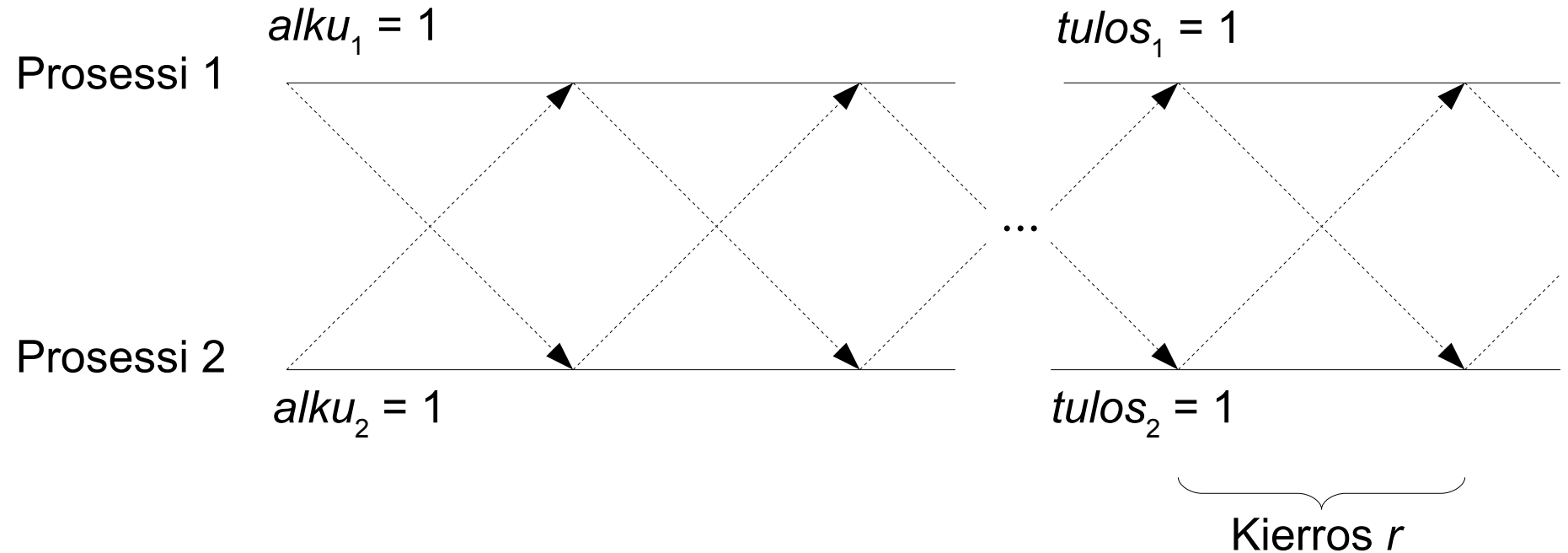
Ratkeamattomuustodistus: suoritus α



- Suoritus etenee kuvassa vasemmalta oikealle
- Nuolet kuvaavat onnistuneesti toimitettuja viestejä



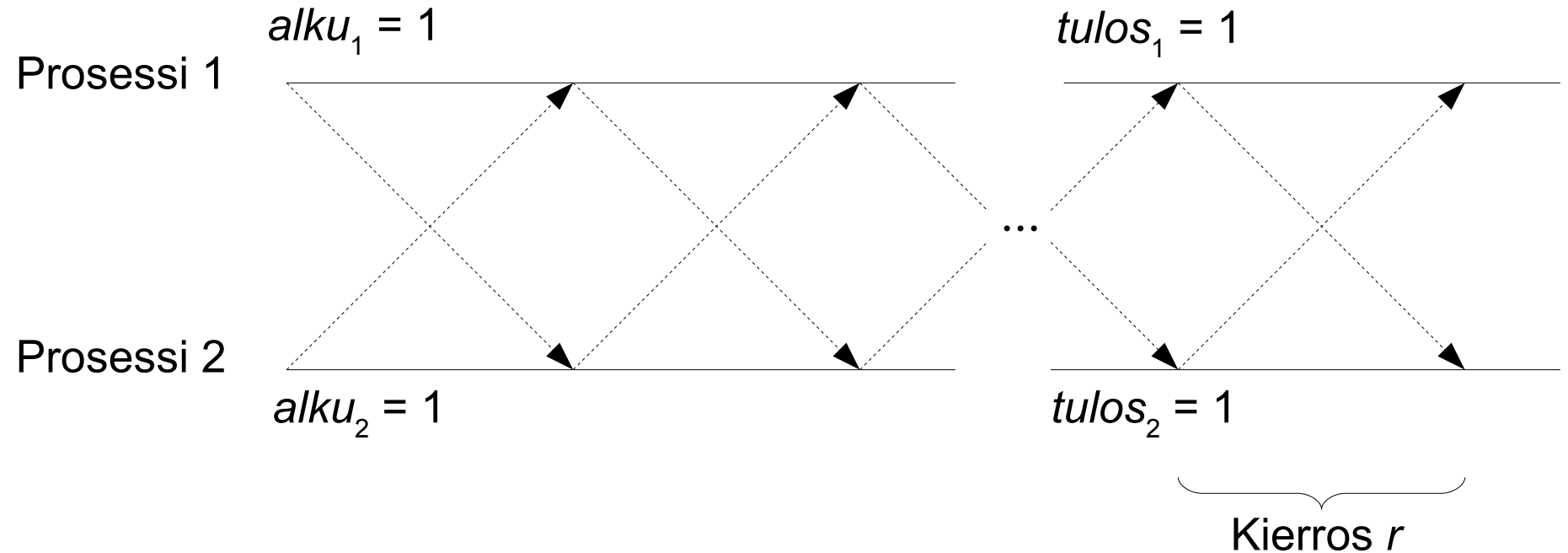
Ratkeamattomuustodistus: suoritus α



- Kiinnitetään r siten, että molemmat prosessit päättävät r kierroksen kuluessa
 - Vastaoletuksen ja ehdon (a) mukaan r on olemassa



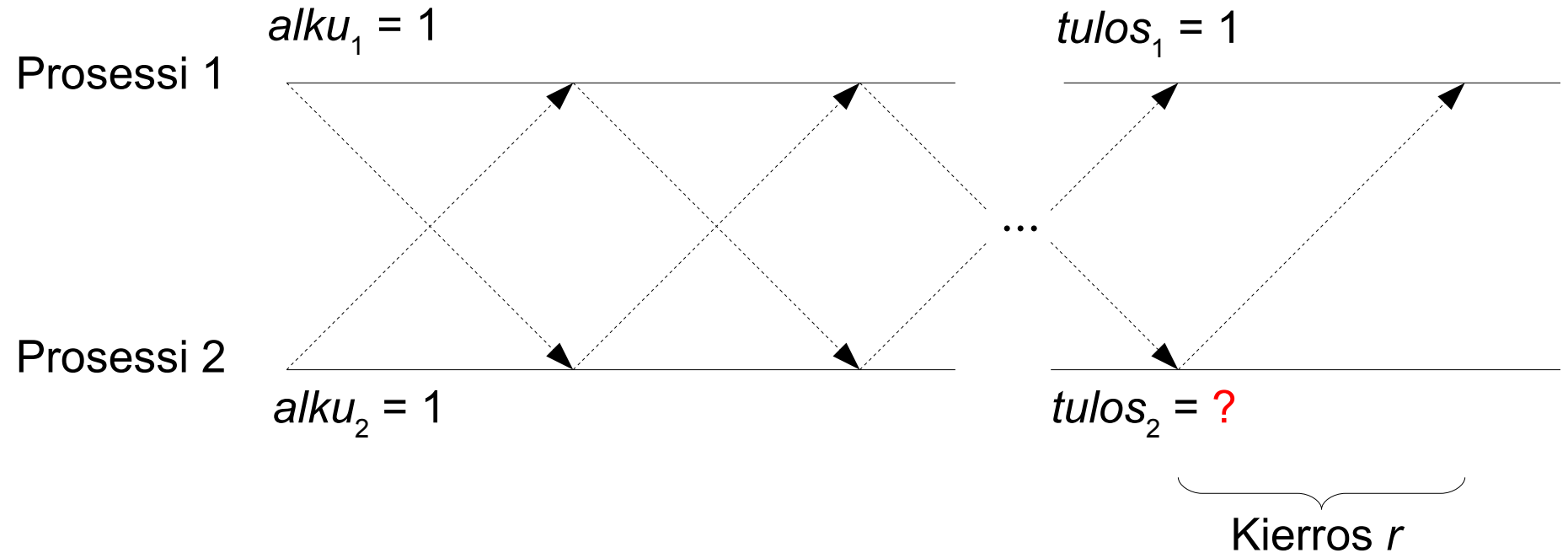
Ratkeamattomuustodistus: suoritus α_1



- Olkoon suoritus α_1 muuten sama kuin suoritus α , mutta kierroksen r jälkeen kaikki viestit katoavat
 - Tulokset eivät muutu, koska päätökset tehtiin r kierroksen aikana



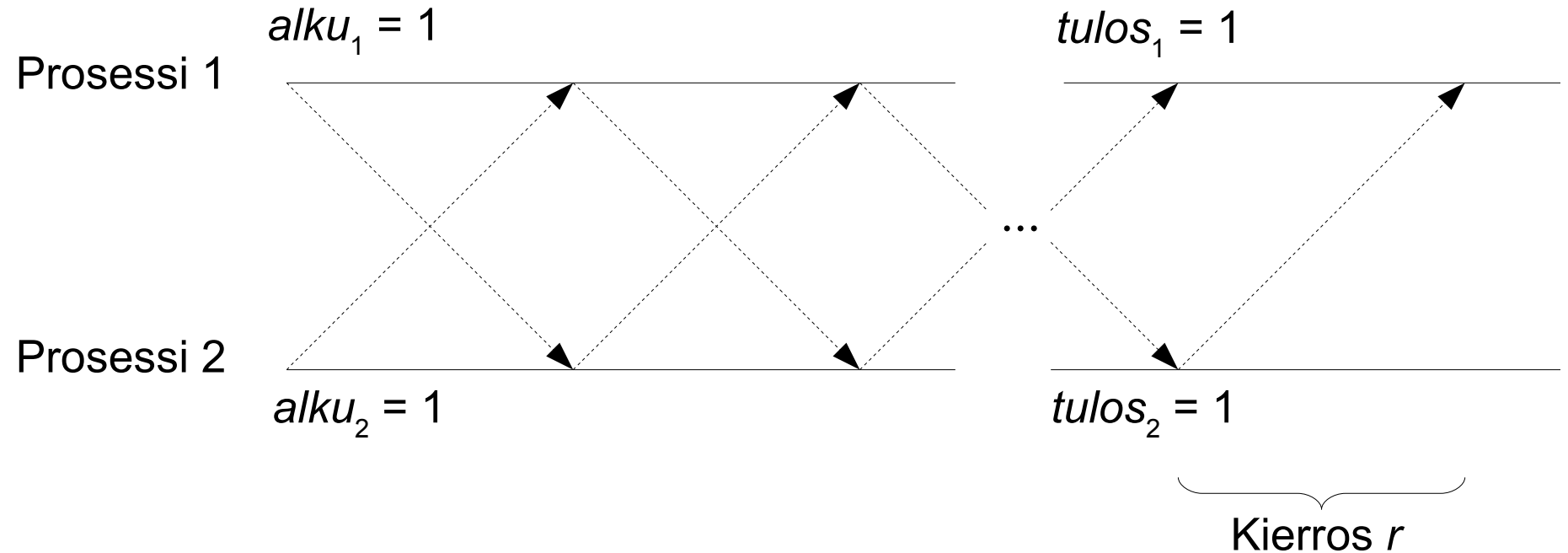
Ratkeamattomuustodistus: suoritus α_2



- Olkoon suoritus α_2 muuten sama kuin suoritus α_1 , mutta prosessin 1 kierroksella r lähettämä viesti katoaa
 - $tulos_1$ ei muutu, koska prosessi 1 ei tiedä viestinsä kadonneen
 - Suoritukset α_1 ja α_2 ovat prosessin 1 kannalta samat!



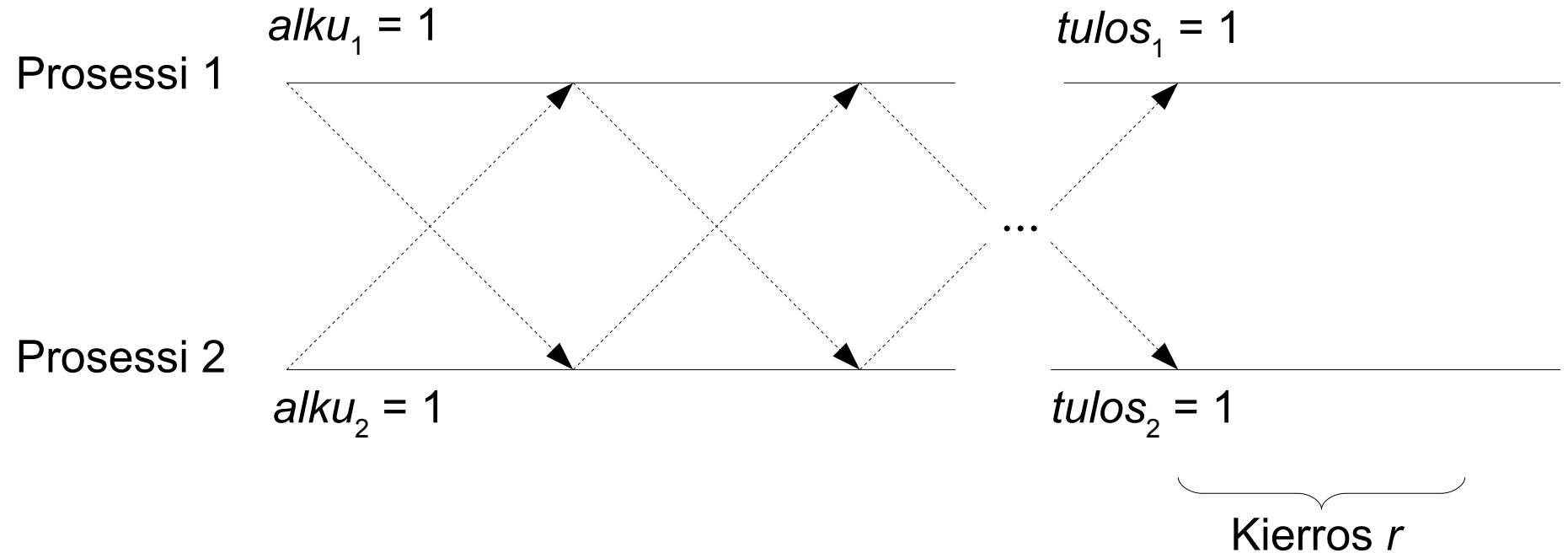
Ratkeamattomuustodistus: suoritus α_2



- Prosessin 2 kannalta suoritukset α_2 ja α_1 ovat erilaisia
 - Prosessi 2 ei vastaanota kierroksen r viestiä
 - Vastaoletuksen ja ehdon (b) perusteella kuitenkin $tulos_2 = 1$



Ratkeamattomuustodistus: suoritus α_3



- Olkoon suoritus α_3 muuten sama kuin suoritus α_2 , mutta prosessin 2 kierroksella r lähettämä viesti katoaa
 - $tulos_2 = 1$, koska prosessi 2 ei tiedä viestinsä kadonneen
 - $tulos_1 = 1$ vasta oletuksen ja ehdon (b) perusteella

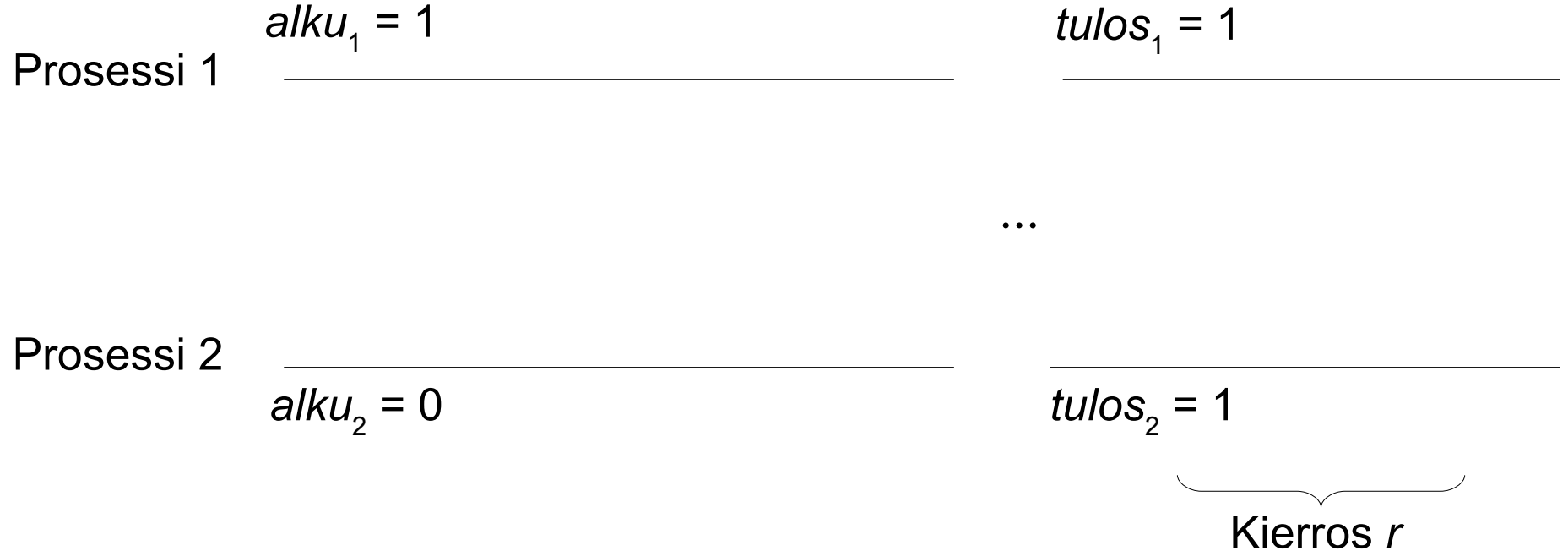


Ratkeamattomuustodistus: suoritus α_3 vs. α_1

- Ainoa ero suoritusten α_3 ja α_1 välillä on viestien katoaminen kierroksella r
 - Ei vaikuta lopputulokseen: molemmissa $tulos_1 = tulos_2 = 1$
- Jatkamalla edellä kuvatulla tavalla prosessien väliset viestit voidaan kadottaa jokaisella kierroksella $r - 1, r - 2, \dots$
- Päädytään suoritukseen α' , jossa prosessit eivät vastaanota yhtäkään viestiä
 - Vastaoletuksen ja ehtojen mukaan edelleen
$$tulos_1 = tulos_2 = 1$$



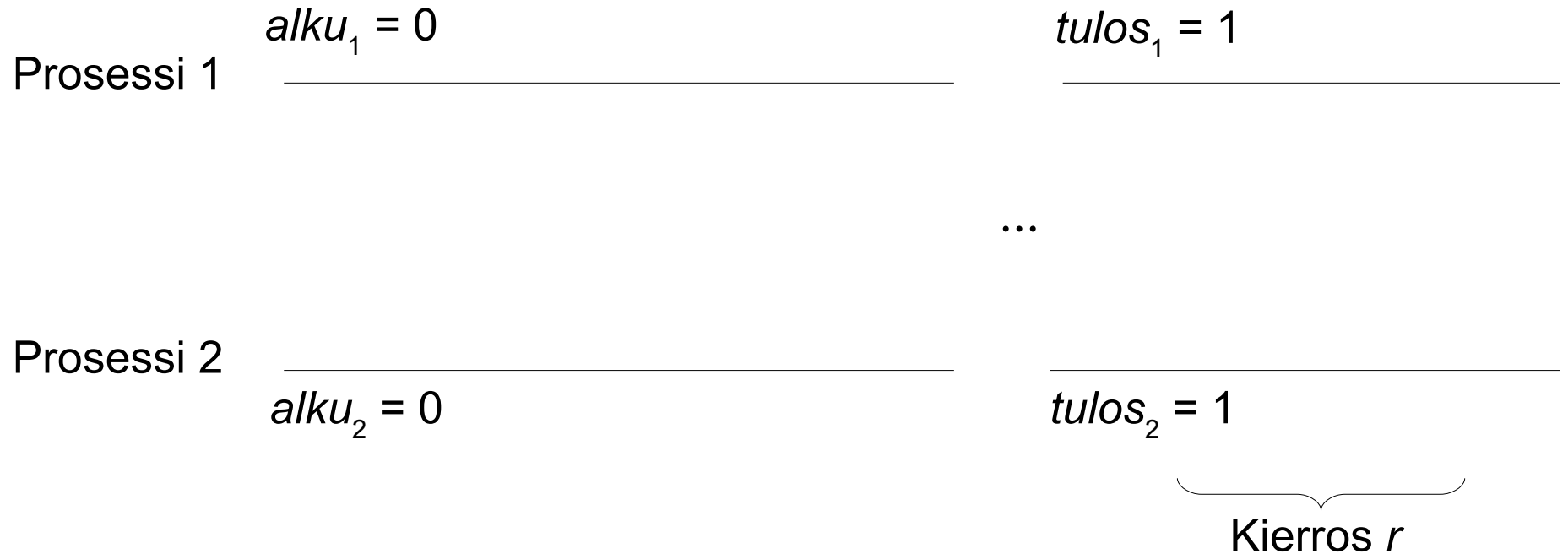
Ratkeamattomuustodistus: suoritus α''



- Olkoon suoritus α'' muuten sama kuin suoritus α' , mutta prosessin 2 alkutila on $alku_2 = 0$
 - $tulos_1 = 1$, koska prosessin 1 kannalta α'' ja α' ovat samat
 - $tulos_2 = 1$ vastaoletuksen ja ehdon (b) perusteella



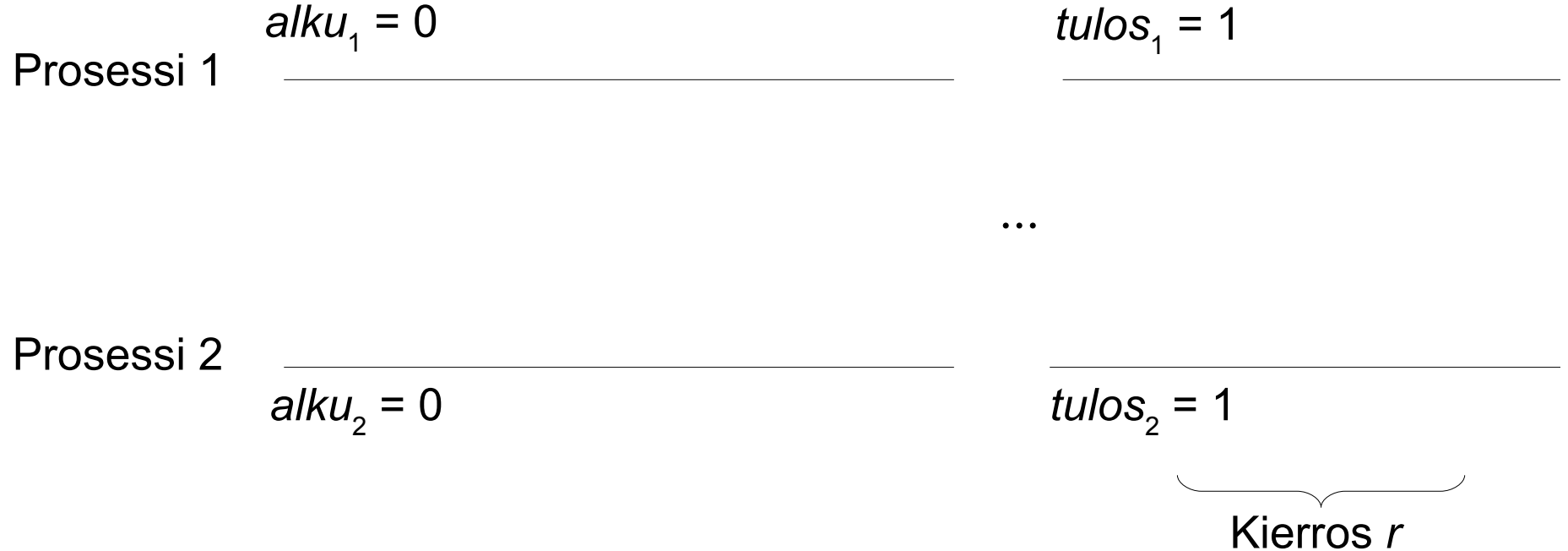
Ratkeamattomuustodistus: suoritus α'''



- Olkoon suoritus α''' muuten sama kuin suoritus α'' , mutta prosessin 1 alkutila on $alku_1 = 0$
 - $tulos_2 = 1$, koska prosessin 2 kannalta α''' ja α'' ovat samat
 - $tulos_1 = 1$ vastaoletuksen ja ehdon (b) perusteella



Ratkeamattomuustodistus: suoritus α'''



- Suorituksesta α''' seuraa ristiriita ehdon (c) kanssa:
 - $alku_1 = alku_2 = 0$, mutta silti $tulos_1 = tulos_2 = 1$
- Konsensusongelma ei ratkea deterministisellä algoritmilla



Satunnaistettu algoritmi

- Ratkaisee konsensusongelman, kun sallitaan *virhetilanne* todennäköisyydellä ε
 - Virhetilanne: päätös ei ole yksimielinen ($tulos_i \neq tulos_j$)
- Algoritmin parametrina kierrosten lukumäärä r
 - Prosessit tekevät päätöksen kierroksella r
 - Todennäköisyys ε riippuu r :n valinnasta
- *Vastustaja* (adversary) yrittää hankaloittaa tilannetta valitsemalla
 - Alkutilat $alku_i$ jokaiselle i
 - Prosessien välillä onnistuneesti toimitetut viestit



Satunnaistettu algoritmi: ehdot

- Määritellään seuraavat ehdot, joiden tulee täytyä:

(a) $Pr^B [\exists i, j \in [1, n] : tulos_i = 0 \wedge tulos_j = 1] \leq \varepsilon$

(b) Kaikkien alkutilojen ollessa 0 myös kaikki $tulos_i = 0$

(c) Kaikkien alkutilojen ollessa 1 myös kaikki $tulos_i = 1$,
jos kaikki viestit on toimitettu onnistuneesti

- Pr^B on todennäköisyys vastustajalla B

- Ehto (a) tulee olla voimassa mille tahansa vastustajalle B



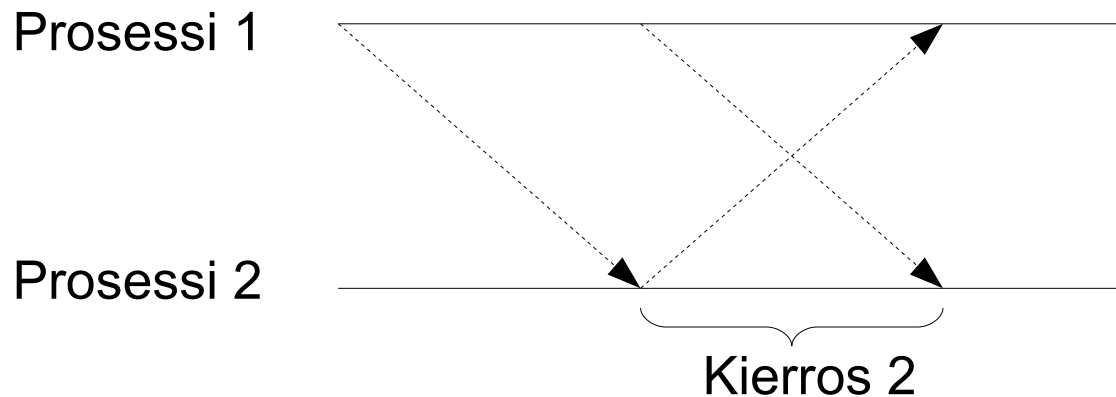
Satunnaistettu algoritmi: määritelmiä

- *Viestinvälitystä* (communication pattern) kuvaa osajoukko

$$\gamma \subseteq \{ (i, j, k) \mid i, j \in [1, n] \wedge 1 \leq k \leq r \}$$

- $(i, j, k) \in \gamma$, jos ja vain jos prosessi j vastaanotti prosessin i kierroksella k lähettämän viestin

- Esimerkiksi: $\gamma = \{ (1, 2, 1), (1, 2, 2), (2, 1, 2) \}$





Satunnaistettu algoritmi: informaation virtaus

■ Määritellään järjestys \leq_γ seuraavasti

1. $(i, k) \leq_\gamma (i, k')$ kaikille $i \in [1, n]$ ja kaikille $0 \leq k \leq k'$
2. Jos $(i, j, k) \in \gamma$, niin $(i, k-1) \leq_\gamma (j, k)$
3. Jos $(i, k) \leq_\gamma (i', k')$ ja $(i', k') \leq_\gamma (i'', k'')$,
niin myös $(i, k) \leq_\gamma (i'', k'')$

■ Järjestys on refleksiivinen ja transitiivinen (kohdat 1 ja 3)

■ Kohta 2 kuvaa informaation kulkua:

- Prosessi j vastaanottaa prosessin i kierroksen k viestin



Satunnaistettu algoritmi: informaatiomäärä

- *Informaatiomäärä* (information level) prosessille i kierroksella $0 \leq k \leq r$ on $taso_\gamma(i, k)$:
 1. Jos $k=0$, niin $taso_\gamma(i, k)=0$
 2. Jos $k \neq 0$ ja on olemassa $j \neq i$ s.e. $(j, 0) \not\leq_\gamma(i, k)$,
niin $taso_\gamma(i, k)=0$
- Prosessin informaatiomäärä on aluksi nolla
- Kohdan 2 mukaan prosessin informaatiomäärä on nolla kunnes se saa tiedon kaikista muista prosesseista
 - Informaatio voi kulkeutua useamman prosessin läpi, koska järjestys \leq_γ on transitiivinen



Satunnaistettu algoritmi: informaatiomäärä

- *Informaatiomäärä* (information level) prosessille i kierroksella $0 \leq k \leq r$ on $taso_\gamma(i, k)$:

3. Jos $k \neq 0$ ja $(j, 0) \leq_\gamma(i, k)$ kaikille $j \neq i$,

niin $taso_\gamma(i, k) = 1 + \min \{ \ell_j \mid j \neq i \}$,

missä $\ell_j = \max \{ taso_\gamma(j, k') \mid (j, k') \leq_\gamma(i, k) \}$

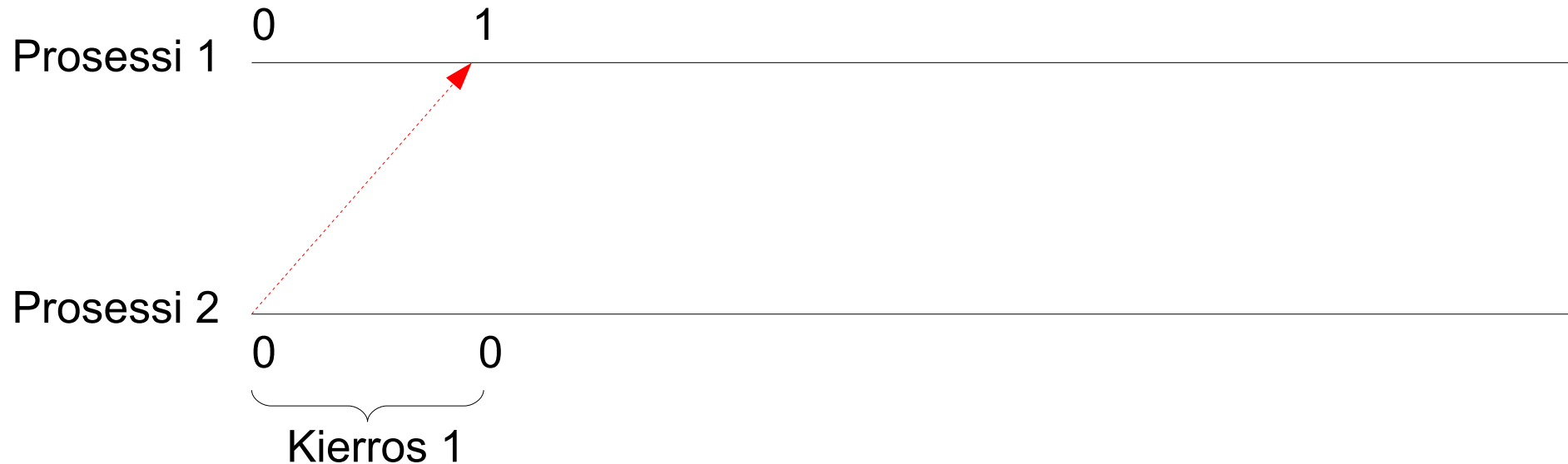
- ℓ_j on suurin informaatiomäärä, jonka prosessi i tietää prosessin j saavuttaneen
- Toisin sanoen:
 - Kun prosessi i saa tietää, että kaikki muut prosessit ovat saavuttaneen informaatiomäärän t , prosessin i informaatiomäärä on $t + 1$



Satunnaistettu algoritmi: informaatiomäärä

■ Esimerkki:

$$\gamma = \{ (2, 1, 1), (2, 1, 2), (1, 2, 3), (2, 1, 3), \\ (1, 2, 4), (2, 1, 4), (2, 1, 5), (1, 2, 6) \}$$

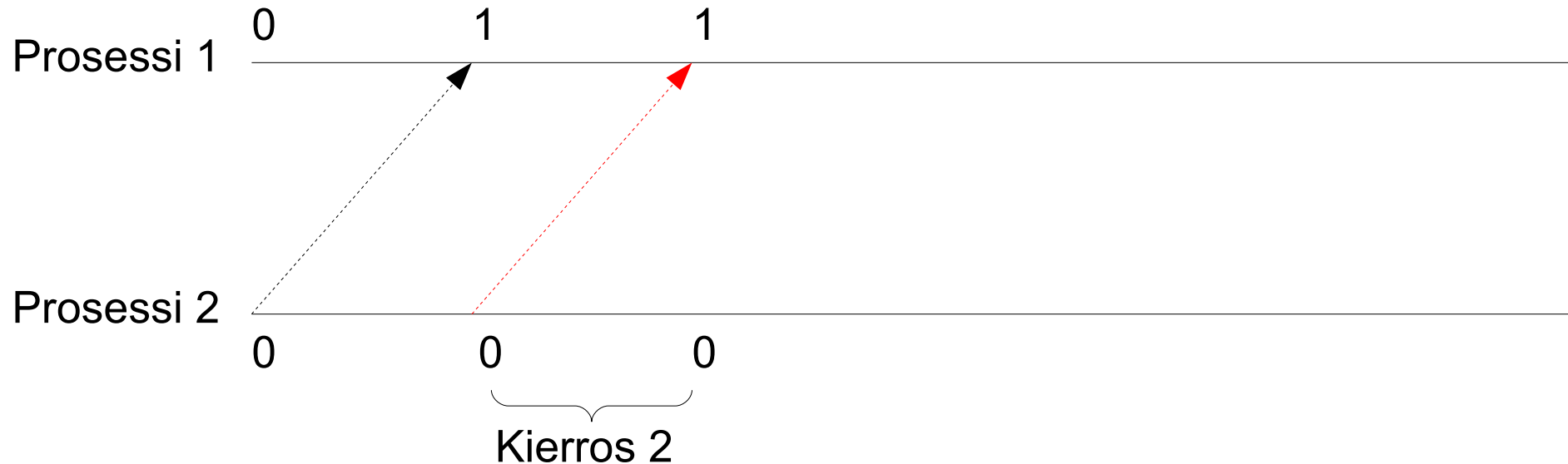




Satunnaistettu algoritmi: informaatiomäärä

■ Esimerkki:

$$\gamma = \{ (2, 1, 1), (2, 1, 2), (1, 2, 3), (2, 1, 3), \\ (1, 2, 4), (2, 1, 4), (2, 1, 5), (1, 2, 6) \}$$

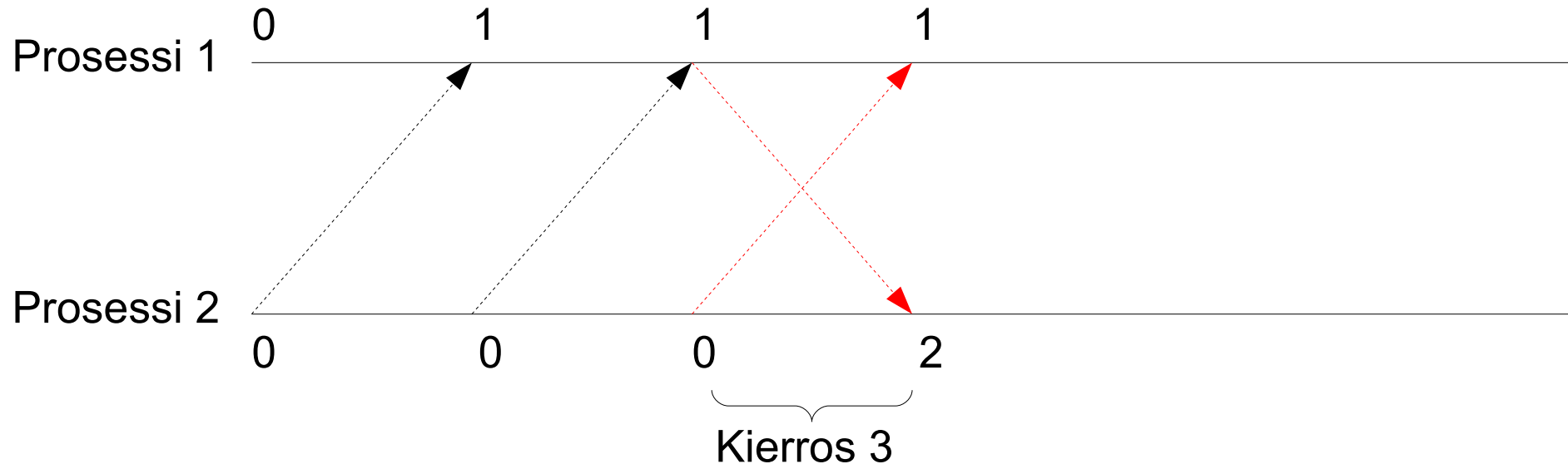




Satunnaistettu algoritmi: informaatiomäärä

■ Esimerkki:

$$\gamma = \{ (2, 1, 1), (2, 1, 2), (1, 2, 3), (2, 1, 3), \\ (1, 2, 4), (2, 1, 4), (2, 1, 5), (1, 2, 6) \}$$

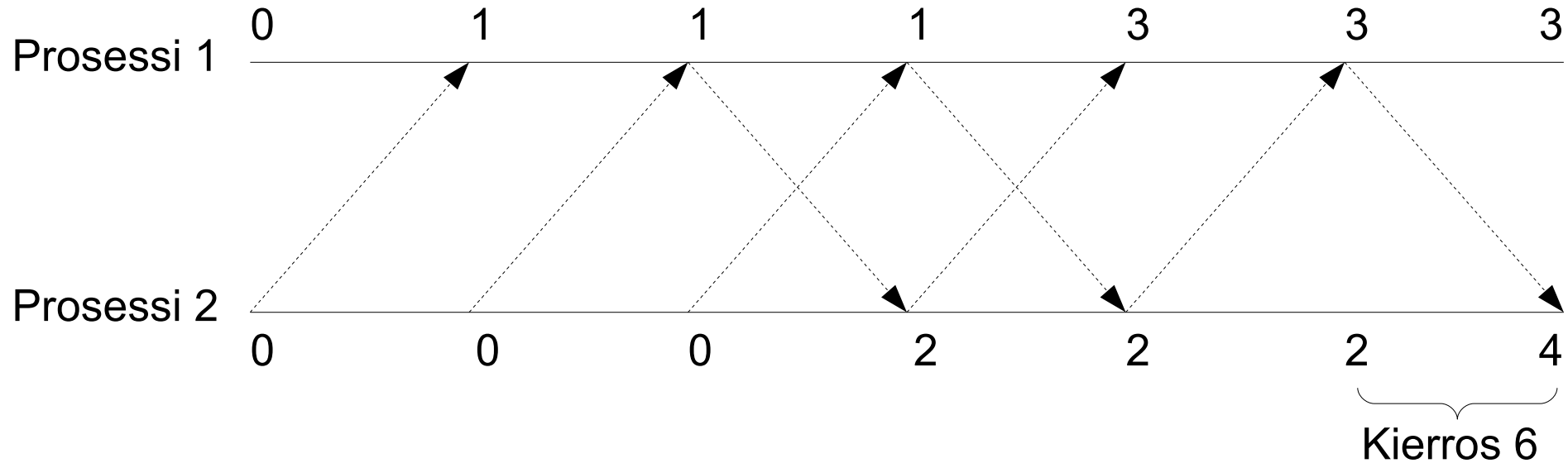




Satunnaistettu algoritmi: informaatiomäärä

■ Esimerkki:

$$\gamma = \{ (2, 1, 1), (2, 1, 2), (1, 2, 3), (2, 1, 3), \\ (1, 2, 4), (2, 1, 4), (2, 1, 5), (1, 2, 6) \}$$





Satunnaistettu algoritmi

- Informaatiomäärän avulla voidaan vihdoin määritellä satunnaistettu algoritmi konsensusongelmaan:
- Kiinnitetään kierrosten lukumäärä r
- Prosessi 1 arpoo *raja-arvon* väliltä $[1, r]$
 - Aluksi raja-arvo on vain prosessin 1 tiedossa
- Jokainen prosessi pitää kirjaa omasta informaatiomäärästään r kierroksen ajan
 - Vaatii, että seurataan kaikkien prosessien informaatiomääriä
- ...sekä tiedossa olevista alkutiloista $alku_j$, kaikille j



Satunnaistettu algoritmi: viestien sisältö

- Prosessien lähettämät viestit sisältävät:
 - Oman, edellisen kierroksen informaatiomäärän $taso_{\gamma}(i, k - 1)$
 - Kaikkien muiden prosessien suurimman tiedossa olevan informaatiomäärän
 - Kaikki tiedossa olevat alkutilat $alku_i$
 - Raja-arvon, jos se on prosessin tiedossa
- Viesti lähetetään verkon kaikille muille prosesseille jokaisella kierroksella $0 \leq k \leq r$



Satunnaistettu algoritmi: päätöksen tekeminen

- Kierroksen r lopuksi valitaan $tulos_i \in \{0, 1\}$
 - Jos $taso_\gamma(i, r) = 0$, valitaan aina $tulos_i = 0$
 - Jos $taso_\gamma(i, r) > 0$, niin kaikki alkutilat $alku_j$ ja raja-arvo ovat prosessin i tiedossa
- Tarkistetaan ylittääkö oma informaatiomäärä $taso_\gamma(i, r)$ prosessin 1 arpoman raja-arvon väliltä $[1, r]$
 - Jos ylittää ja kaikki $alku_j = 1$, niin valitaan $tulos_i = 1$
 - Muutoin aina $tulos_i = 0$
- Toisin sanoen:
 - Jos yksikin $alku_j = 0$, valitaan aina $tulos_i = 0$



Satunnaistettu algoritmi: viestin vastaanotto

■ Aluksi $taso_i[j] = -1$ kaikilla $j \neq i$

Algoritmi $saapuva_i(L, V, k)$

1 **if** $k \neq tuntematon$ **then** $raja_i \leftarrow k$

2 **for all** $j \neq i$ **do**

3 **if** $V[j] \neq tuntematon$ **then** $alku_i[j] \leftarrow V[j]$

4 **if** $L[j] > taso_i[j]$ **then** $taso_i[j] \leftarrow L[j]$

5 $taso_i \leftarrow 1 + \min \{ taso_i[j] \mid j \neq i \}$

6 **if** $kierros = r$ **then**

7 **if** $taso_i \geq raja_i$ **and** $alku_i[j] = 1$ kaikille j **then**

$tulos_i \leftarrow 1$

8 **else** $tulos_i \leftarrow 0$



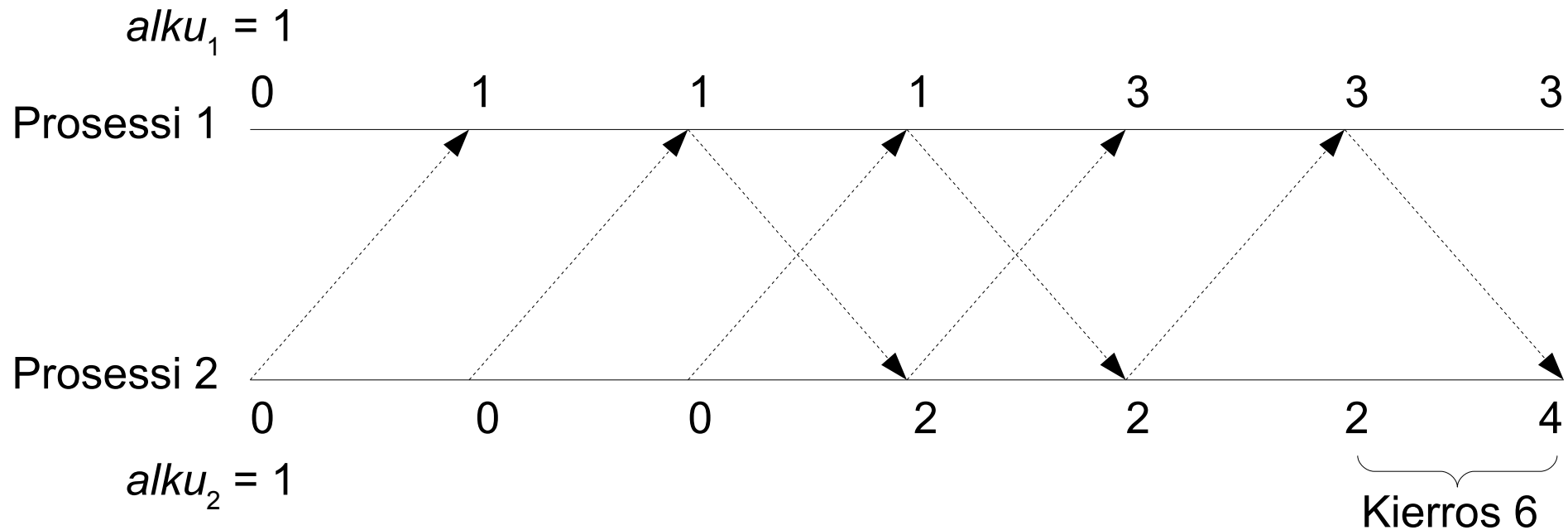
Satunnaistettu algoritmi: yhteenveto

- Prosessi 1 arpoo raja-arvon väliltä $[1, r]$
 - Ainut epädeterministinen osa algoritmossa!
 - Muu suoritus riippuu vastustajasta B
- Kierroksella r tarkistetaan onko prosessin informaatiomäärä vähintään yhtäsuuri kuin raja-arvo
 - Jos lisäksi kaikki $alku_j = 1$, niin valitaan $tulos_i = 1$
 - Muutoin aina $tulos_i = 0$
- Osoittautuu, että konsensusehto rikkoutuu korkeintaan todennäköisyydellä $\varepsilon = \frac{1}{r}$
 - Pätee kaikille vastustajille B



Satunnaistettu algoritmi: esimerkki

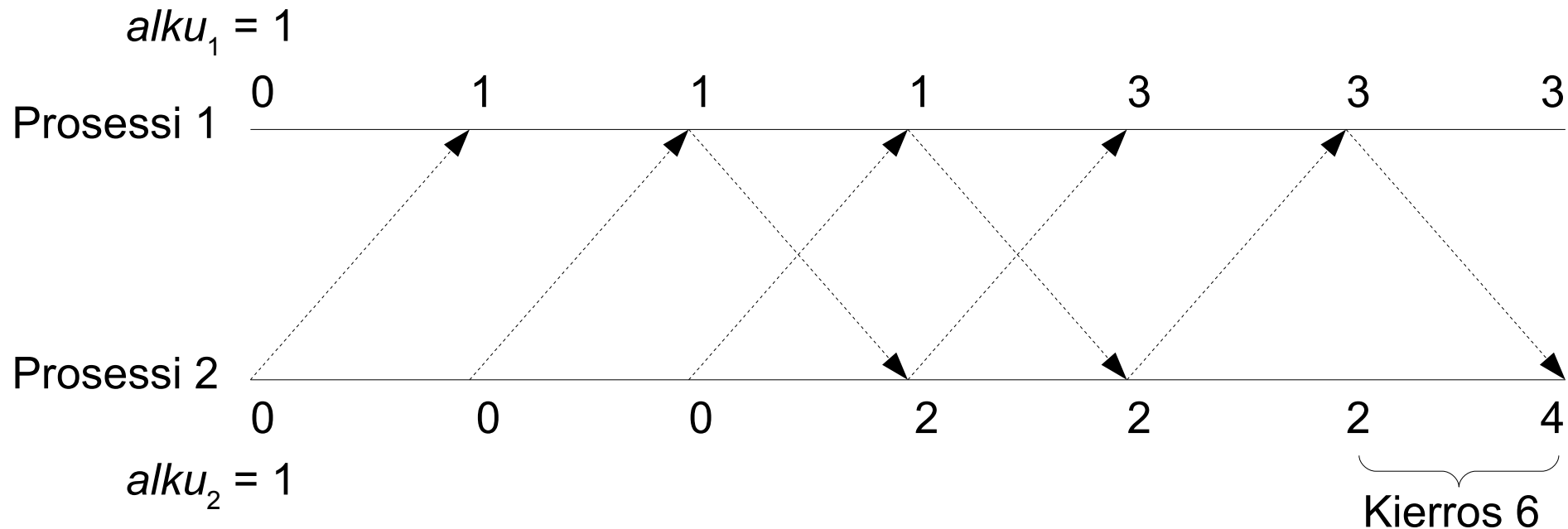
- Tarkastellaan tilannetta kierroksen $r = 6$ päättyessä
 - Jos raja-arvo on pienempi kuin 4, niin $tulos_1 = tulos_2 = 1$ 😊





Satunnaistettu algoritmi: esimerkki

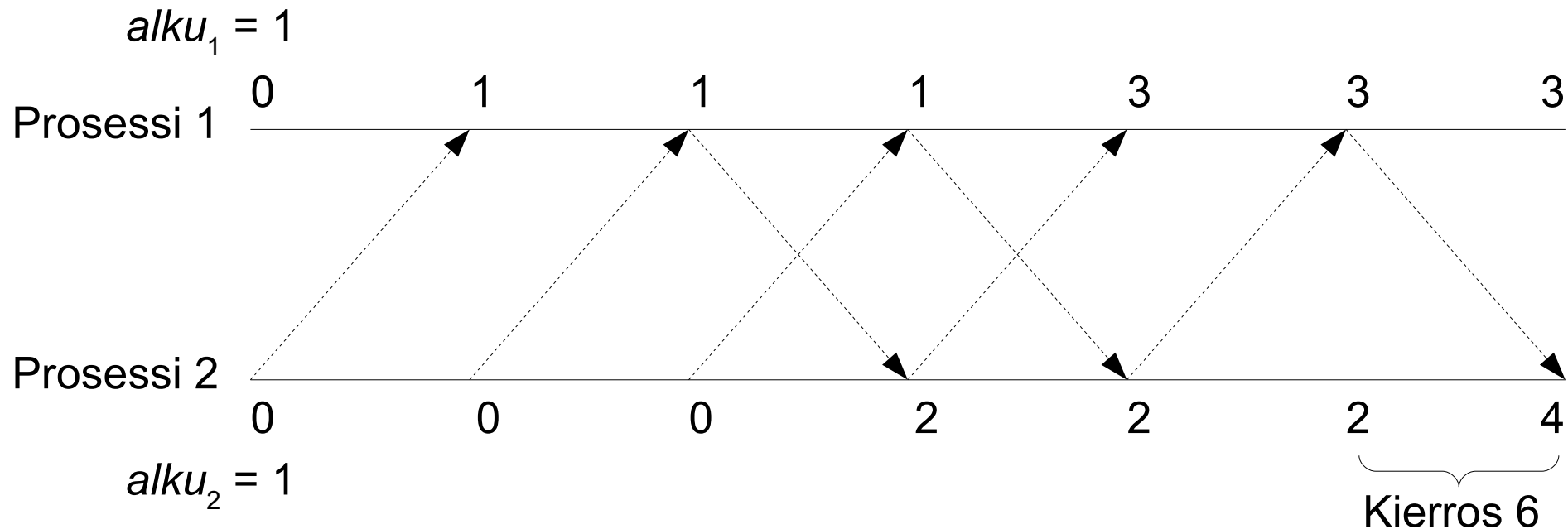
- Tarkastellaan tilannetta kierroksen $r = 6$ päättyessä
 - Jos raja-arvo on pienempi kuin 4, niin $tulos_1 = tulos_2 = 1$ 😊
 - Jos raja-arvo on suurempi kuin 4, niin $tulos_1 = tulos_2 = 0$ 😊





Satunnaistettu algoritmi: esimerkki

- Tarkastellaan tilannetta kierroksen $r = 6$ päättyessä
 - Jos raja-arvo on pienempi kuin 4, niin $tulos_1 = tulos_2 = 1$ 😊
 - Jos raja-arvo on suurempi kuin 4, niin $tulos_1 = tulos_2 = 0$ 😊
 - Jos raja-arvo on 4, niin $tulos_1 = 0$ ja $tulos_2 = 1$ ❌
 - Todennäköisyys $\varepsilon = \frac{1}{6}$, muilla alkutiloilla todennäköisyys 0





Satunnaistettu algoritmi: teoreettinen alaraja

- Kuinka hyvä tulos on kyseessä?

$$Pr^B [\exists i, j \in [1, n] : tulos_i = 0 \wedge tulos_j = 1] \leq \frac{1}{r}$$

- Mikä tahansa r kierrosta toimiva satunnaistettu algoritmi rikkoo konsensususehtoa vähintään todennäköisyydellä $\frac{1}{r+1}$



Satunnaistettu algoritmi: tiukennetut ehdot

■ Myös seuraavat, tiukemmat ehdot pätevät:

(a) Jos jokin alkutiloista on 0, kaikki prosessit valitsevat $tulos_i = 0$

(b) Kaikkien alkutilojen ollessa 1 pätee kaikilla B

$$Pr^B [\text{kaikki prosessit valitsevat } tulos_i = 1] \geq \frac{\ell}{r}$$

missä ℓ on pienin informaatiomäärä kierroksella r

■ Esimerkiksi yhden viestin kadotessa todennäköisyys Pr^B on vähintään $\frac{r-1}{r}$



Konsensusongelma: yhteenveto

- Konsensusongelma hajautetussa järjestelmässä
 - Useita variaatioita
 - Tarkasteltiin tilannetta, jossa $alku_i, tulos_i \in \{0, 1\}$ ja prosessien välinen viestintä on epäluotettava
- Deterministinen lähestymistapa ei toimi!
- Satunnaistetulla algoritmilla
 - Virheen todennäköisyys on mahdollista saada lähelle teoreettista alarajaa

Kiitos!