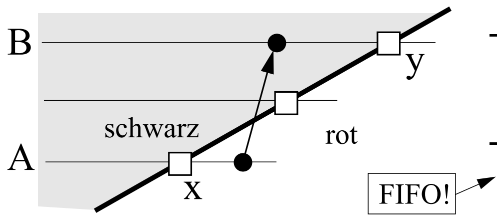
Der Chandy/Lamport-Algorithmus

(erster Schnappschussalgorithmus, 1985 veröffentlicht in ACM TOCS 3, 63-75)

- *Idee:* 1) Setzt FIFO-Kanäle voraus ("flushing-Prinzip")
  Marker schieben In-transit-Nachrichten aus den Kanälen heraus
  2) Flooding als zugrundeliegendes Wellenverfahren

- ¬ ∃ Nachricht aus der Zukunft ==> Schnitt ist *konsistent*
  - vgl. auch frühere Ausführungen zu "virtuell gleichzeitiges Markieren"



- bei x wird ein *Marker* von A nach B gesendet (falls A und B benachbart sind)
- Nachricht kann den Marker nicht überholen ==> nach x gesendete Nachricht kommt *nach* y an

---

- *In-transit-Nachrichten* bei FIFO-Kanälen:
  - Nach der letzten schwarzen Nachricht folgt ein Marker
  - Empfang eines Markers informiert den Empfänger, dass nun über diesen Kanal keine schwarzen Nachrichten mehr ankommen

---

- *Vorteil*: Farben müssen nicht (in Nachrichten) mitgeführt werden
- *Nachteile*:
  - bei dichten Netzen grosse Zahl von Kontrollnachrichten
  - FIFO ist notwendig
  - lokale Zustände müssen i.a. zum Initiator gebracht werden (z.B. mittels Echo-Nachrichten)

Der Chandy/Lamport-Algorithmus (2)

- Globaler Zustand besteht aus den *Prozesszuständen* und allen *Kanalzuständen*

- Im Unterschied zum nicht-FIFO-Fall sind Kanalzustände *Folgen* von Nachrichten, keine Mengen

  Channel State = sequence of messages sent along the channel before the sender's state is recorded, excluding the sequence of messages received along the channel before the receiver's state is recorded

Marker-Sending Rule for a Process p. For each channel *c*, incident on, and directed away from *p*:

p sends one marker along *c* after *p* records its state and before *p* sends further messages along *c*

Marker-Receiving Rule for a Process q. On receiving a marker along channel *c*:

if *q* has not recorded its state **then**
  q records its state;
  q records the state of *c* as the empty sequence
else
  q records the state of *c* as the sequence of messages received along *c* after *q*'s state was recorded and before *q* received the marker along *c*

- Wie verhält sich ein Initiator? Kann der Algorithmus "spontan" von mehreren Prozessen unabhängig voneinander initiiert werden?

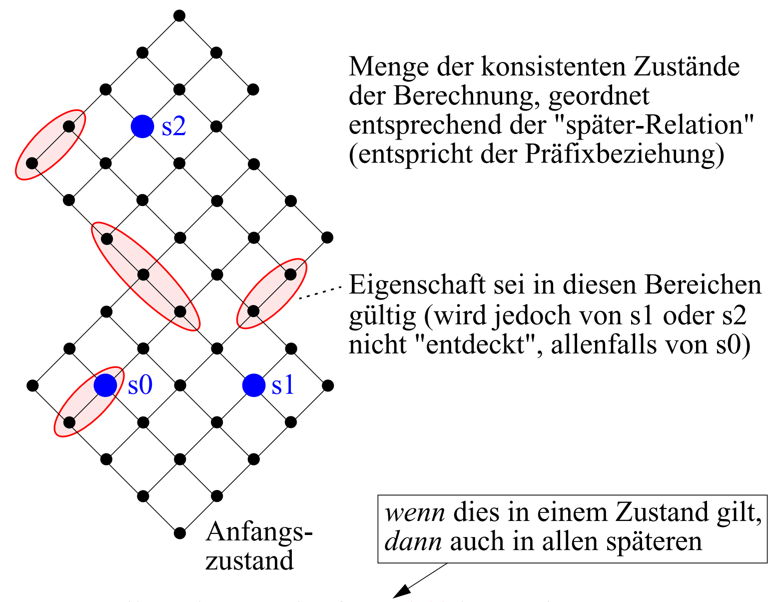
- Wieso ist im ersten Fall der Kanalzustand die leere Folge? Hat jeder so gewonnene globale Zustand einige leere Kanäle?

Türken einen Tag unter "Hausarrest" gestellt

ANKARA (dpa). Mehr als 50 Millionen Einwohner der Türkei standen gestern unter "Hausarrest". Weil die Wählerlisten für ein am 6. September anstehendes Referendum überprüft werden sollten, durfte die Bevölkerung den ganzen Tag die Wohnungen nicht verlassen. In der Volksbefragung soll über Fortdauer oder Aufhebung des seit 1980 bestehenden politischen Betätigungs-Verbots für die früheren Ministerpräsidenten Ecevit und Demirel entschieden werden. Während landesweit zehntausende von Helfern unterwegs waren, um in dem wie ausgestorben wirkenden Land die Eintragungen in die Wählerlisten in den Wohnungen der Wähler zu kontrollieren, attackierten die beiden Oppositionspolitiker die umständliche Methode der Zählung.

Eigenschaften verteilter Berechnungen mit Schnappschüssen "entdecken"?

- Ein wiederholt angewendeter Schnappschussalgorithmus könnte zuerst s1, dann s2 liefern (s2 ist "später" als s1)
  - dazwischen sind Lücken!

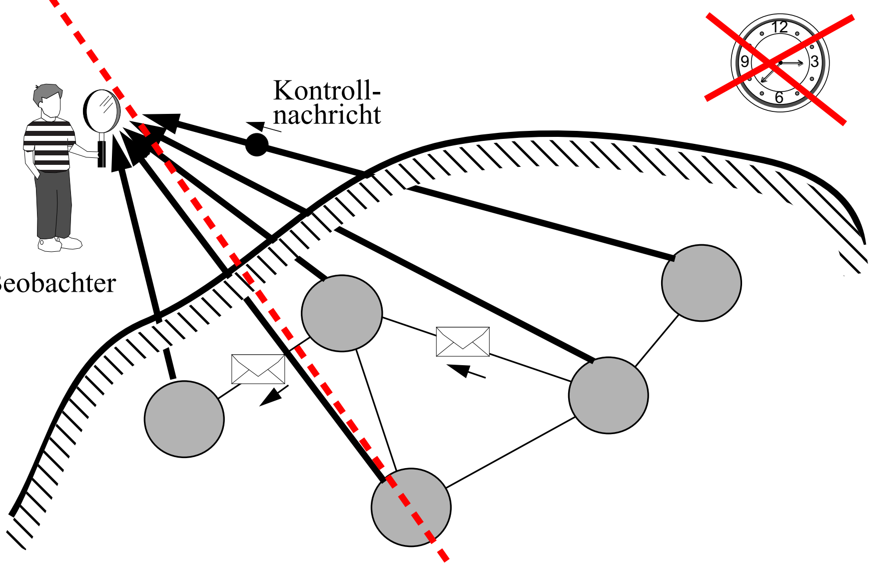


- Sinnvoll, wenn die Eigenschaft stabil ist - aber ansonsten?
  - beachte: wir wissen nicht, ob "in Wirklichkeit" s0 oder s1 eintrat!

- Wir hätten gerne eine lückenlose "Folge" konsistenter Schnappschüsse als eine "*Beobachtung*" der Berechnung
- Allerdings sind Berechnungen nur *halbgeordnete* Mengen (konsistenter) Zustände (also keine Folgen)!

Beobachten verteilter Berechnungen

wie früher schon erwähnt...:

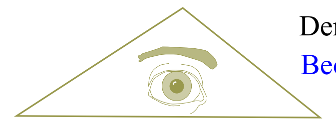


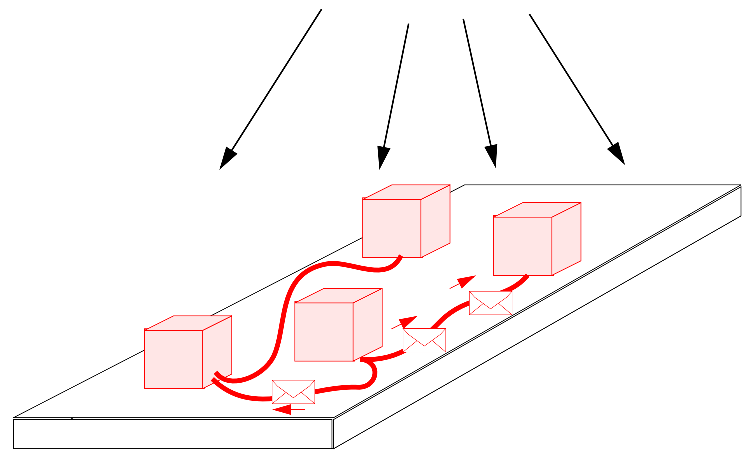
Beobachten geht nur über das Empfangen von "Kontrollnachrichten" (mit unbestimmter Laufzeit)

"Axiom": | Mehrere Prozesse können "niemals" gleichzeitig beobachtet werden |

"Korollar": | Aussagen über den globalen Zustand sind schwierig |

Verteilte Berechnung und Beobachtung

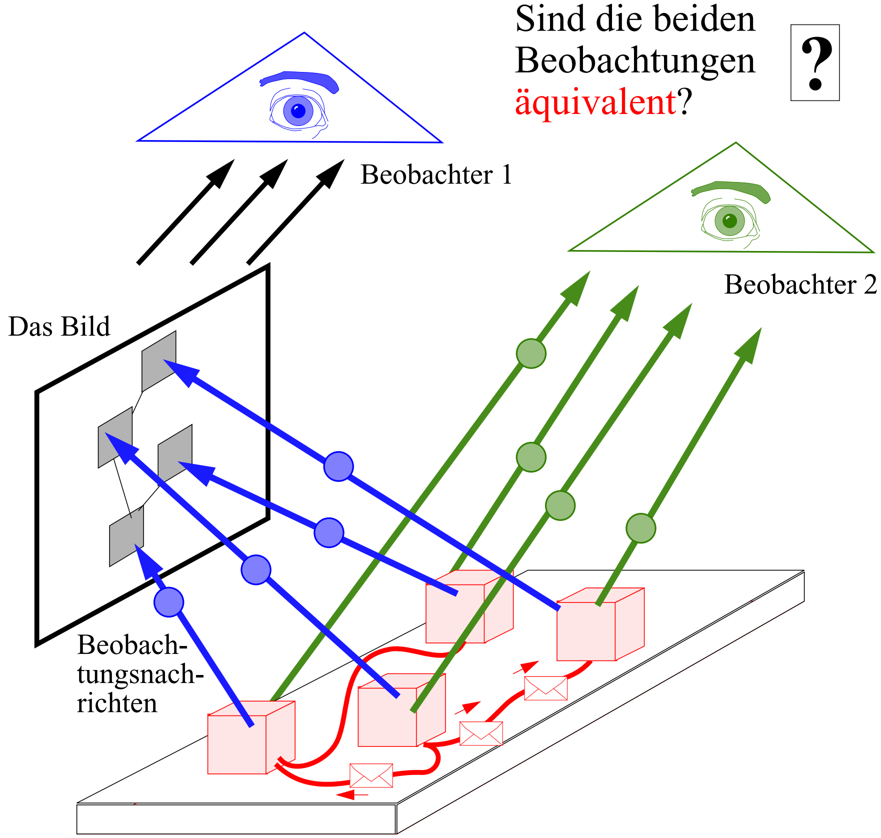




Eine verteilte Berechnung

Beobachtungen...

Sind die beiden Beobachtungen äquivalent?



Probleme:

- Zeitverzögerung der Beobachtung
- Konsistenz des Bildes
- Verzerrung des Verhaltens ("Heisenberg'sche Unschärfe")

Das Paradigma der Beobachtung

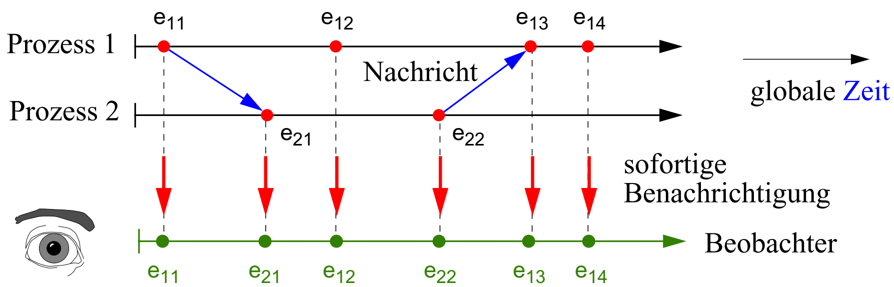
- "Korrektes" Beobachten verteilter Systeme ist ein wichtiges praktisches Problem

- Kausaltreue (auch: "kausal konsistente") Beobachtungen stellen das Kernproblem vieler verteilter Algorithmen dar

==> Wie realisiert man kausaltreue Beobachter?

==> Was sind, formal gesehen, *kausaltreue Beobachter / Beobachtungen ?*

Ideale Beobachtungen



wesentliche Vorkommnisse, z.B.
Nachricht empfangen / senden

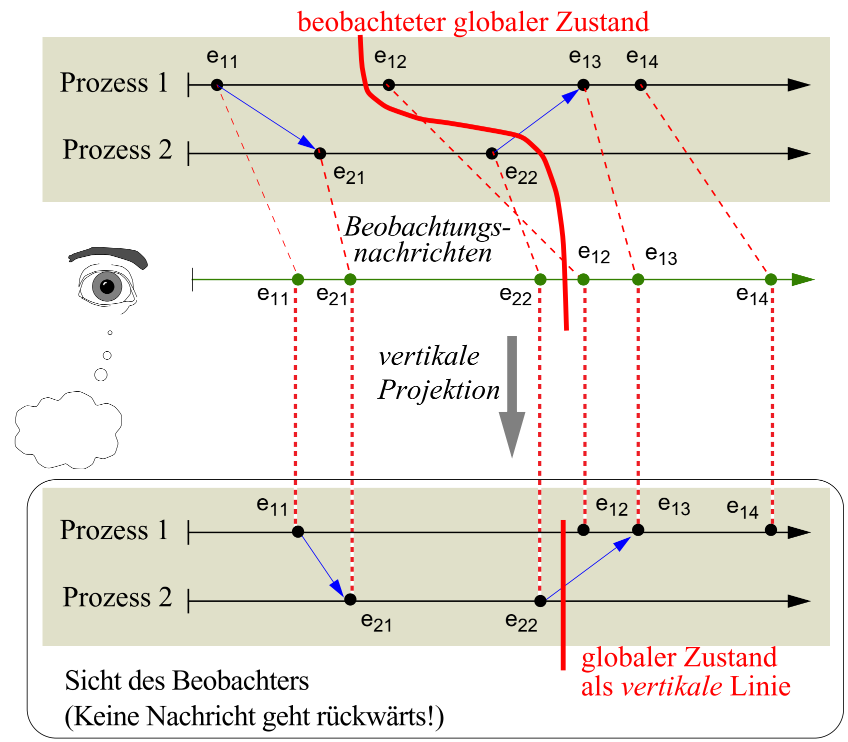
- Beobachtet werden Ereignisse bei den Prozessen
  - Meldung nach aussen über Beobachtungsnachrichten

- Eine solche ideale Beobachtung möchten wir haben, können das aber nicht garantiert bekommen

- Statt dessen könnten wir etwas bekommen, was wir nicht haben wollen...
  (nämlich eine inkonsistente Beobachtung, bei der sich Beobachtungs-nachrichten so überholen, dass Ursache und Wirkung vertauscht werden)

Kausaltreue Beobachtungen

- Ursachen werden stets vor ihren Wirkungen angezeigt
  - dies hoffen wir zu bekommen! (aber wie?)



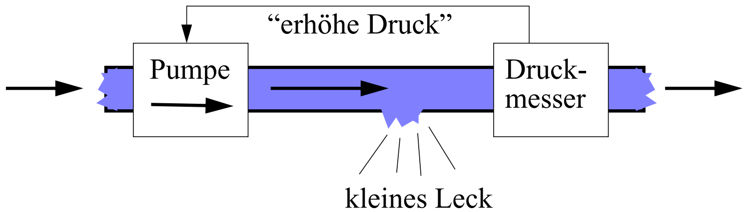
- Beobachtersicht ist nur in irrelevanter Weise verzerrt
  (d.h. ist eine Gummibandtransformation des echten Ablaufes)

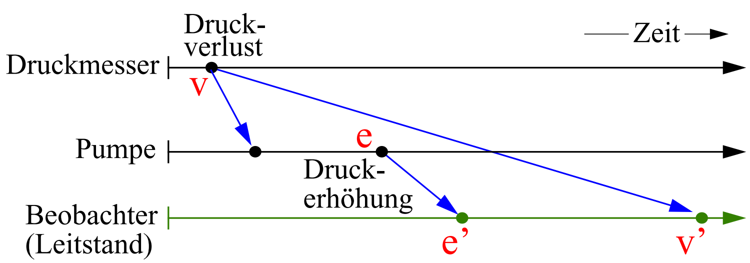
- Beobachteter globaler Zustand ist daher *möglich*

- *gültige*, nicht zu widerlegende Interpretation
- nicht entscheidbar, ob *tatsächlich eingetreten*

Kausal (in)konsistente Beobachtungen

wie früher schon erwähnt...:





Falsche Schlussfolgerung des Beobachters:

Es erhöhte sich der Druck (aufgrund einer unbegründeten Aktivität der Pumpe), es kam zu einem Leck, was durch den abfallenden Druck angezeigt wird.

Forderung (“kausaltreue Beobachtung”):

Ursache stets vor (u.U. indirekter) Wirkung beobachten!

Frage:

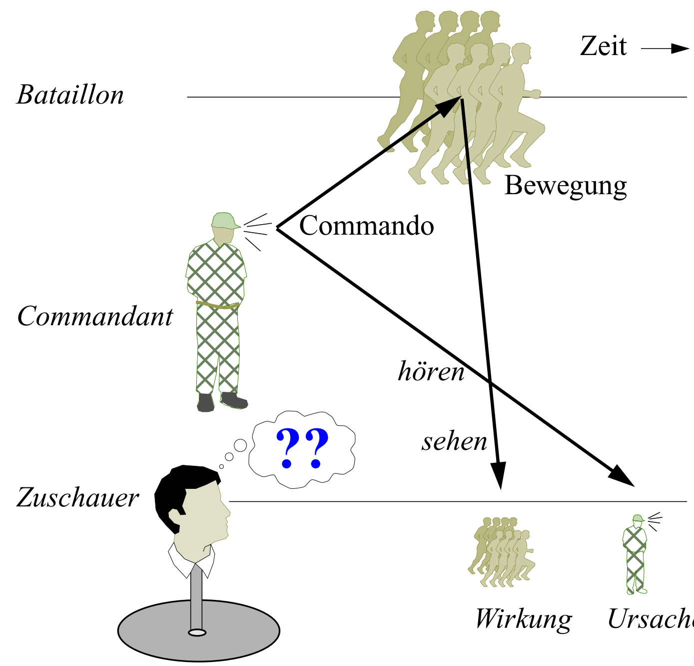
Wie implementiert man kausal konsistente Beobachter?

Kausal (in)konsistente Beobachtungen

Das Beobachtungsproblem ist nicht neu...

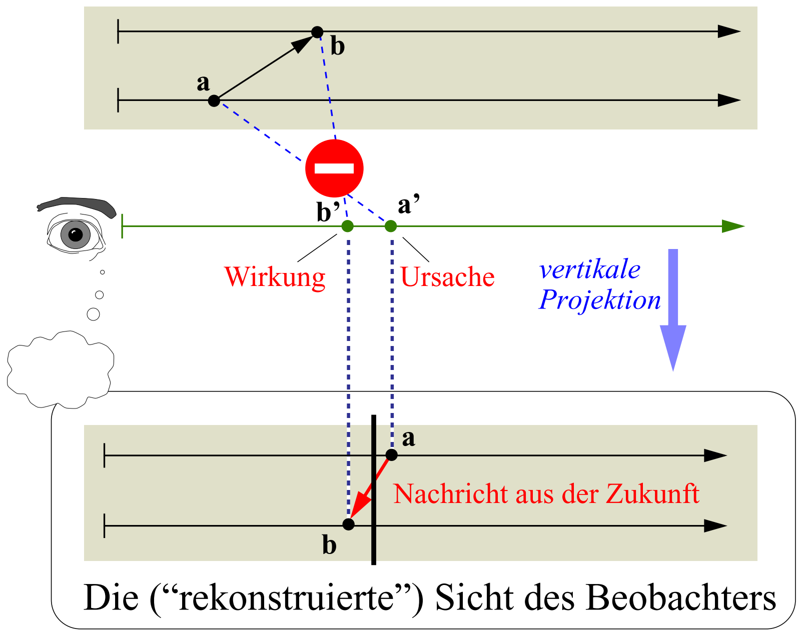
Wenn ein Zuschauer von der Ferne das Exercieren eines Bataillons verfolgt, so sieht er übereinstimmende Bewegungen desselben plötzlich eintreten, *ehe* er die Commandostimme oder das Hornsignal hört; aber aus seiner Kenntnis der *Causalzusammenhänge* weiss er, dass die Bewegungen die *Wirkung* des gehörten Commandos sind, dieses also jenen *objectiv* vorangehen muss, und er wird sich sofort der Täuschung bewusst, die in der Umkehrung der Zeitfolge in seinen Perceptionen liegt.

Christoph von Sigwart (1830-1904) *Logik* (1889)



Kausal inkonsistente Beobachtungen

- Überholungen von Benachrichtigungen:



- In der Interpretation des Beobachters:
  - Nachricht fliesst rückwärts in der Zeit
  - Kausalität ist verletzt (Wirkung vor Ursache!)
  - Beobachteter globaler Zustand nach b ist inkonsistent

- Wir hätten gerne eine kausaltreue Beobachtung, wo die Beobachtersicht nur in irrelevanter Weise verzerrt ist
  - d.h. eine Gummibandtransformation des echten Ablaufes ist (bei der keine Nachricht rückwärts verläuft)
  - wie erzwingt man das?

Kausaltreue Beobachtungen

lineare Erweiterung oder Einbettung

Definition:
Eine *kausaltreue Beobachtung* einer Berechnung ist eine Linearisierung der entsprechenden (partiellen) Kausalordnung (E,<)

Mit anderen Worten:
Jeder “Trace” von Ereignissen, in dem eine Wirkung niemals vor ihrer Ursache erscheint, heisst “kausaltreue Beobachtung”

Bemerkung:
Es gibt i.a. viele unterschiedliche Linearisierungen!

- wieviele? (Grössenordnung in Abhängigkeit der Ereignis- und Prozesszahl?)
- eine sequentielle Berechnung besitzt offenbar nur eine einzige Linearisierung

---

- kausal unabhängige Ereignisse können stets in unterschiedlicher Reihenfolge wahrgenommen werden
- alle kausaltreuen Beobachtungen sind gleichermassen “wahr”
- alle kausaltreuen Beobachter sind sich bzgl. Kausalitätsrelation einig!

Schnitt aller Linearisierungen einer Halbordnung = Halbordnung
(Theorem von Szpilrajn, 1930)

==> der "unstrittige Kern" aller Beobachter ist die Kausalrelation der Berechnung selbst!

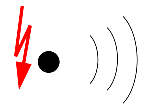
==> in den "wesentlichen" Aspekten stimmen alle Beobachter überein! (die Kausalbeziehung ist also ein beobachterinvariantes, objektives Faktum)

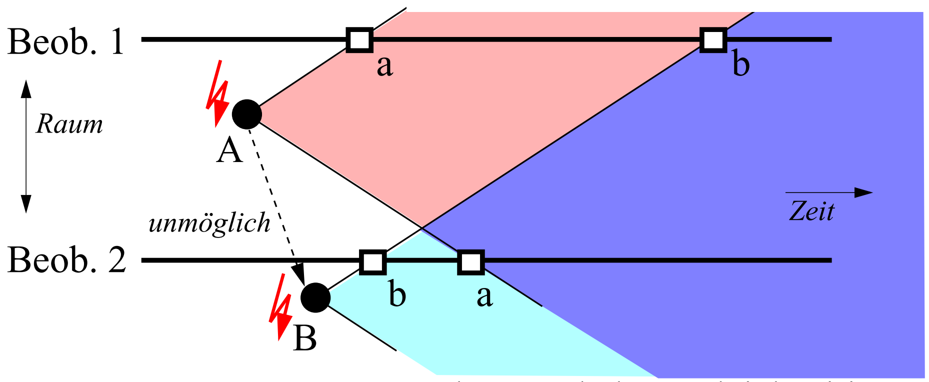
==> vert. Berechnung ist durch die Menge aller ihrer Beobachtungen charakterisiert

Relativierung der Gleichzeitigkeit

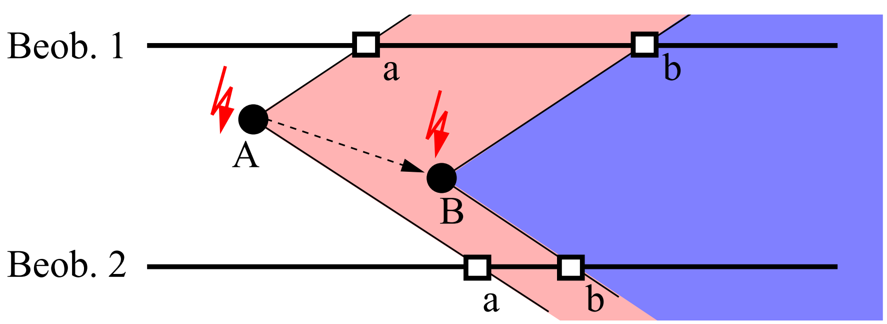
Zwei "kausal unabhängige" Ereigniss können in
beliebiger Reihenfolge beobachtet werden!

Lichtkegel-Prinzip der relativistischen Physik





A und B geschehen "gleichzeitig"
(jedenfalls *kausal unabhängig*!)

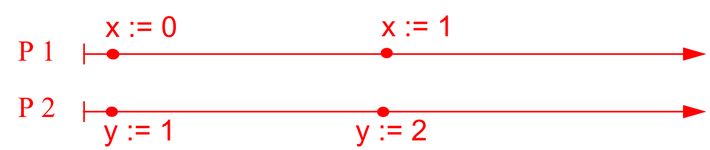


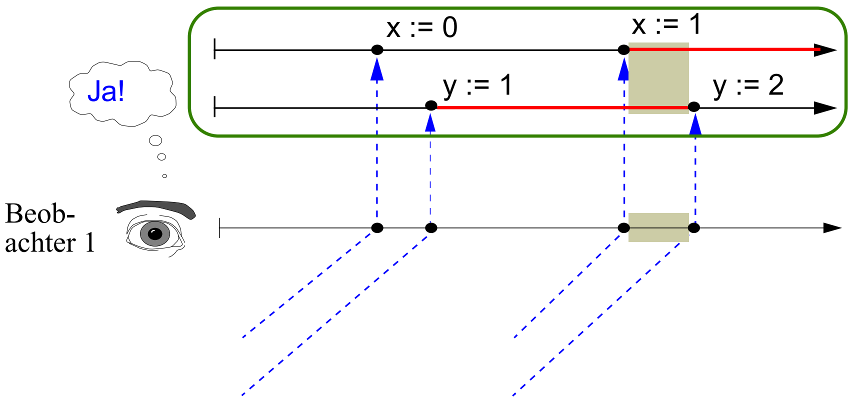
beobachterinvariant
==> *objektive Tatsache*

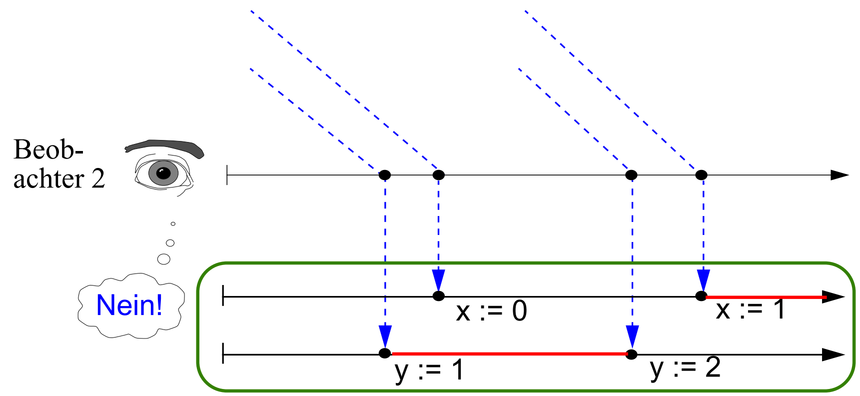
B liegt im Kegel von A -->
B hängt kausal ab von A -->
Alle Beobachter sehen B nach A

Das "Entdecken" globaler Prädikate

Frage: Gilt in dieser Berechnung $\Phi \equiv (x = y)$?





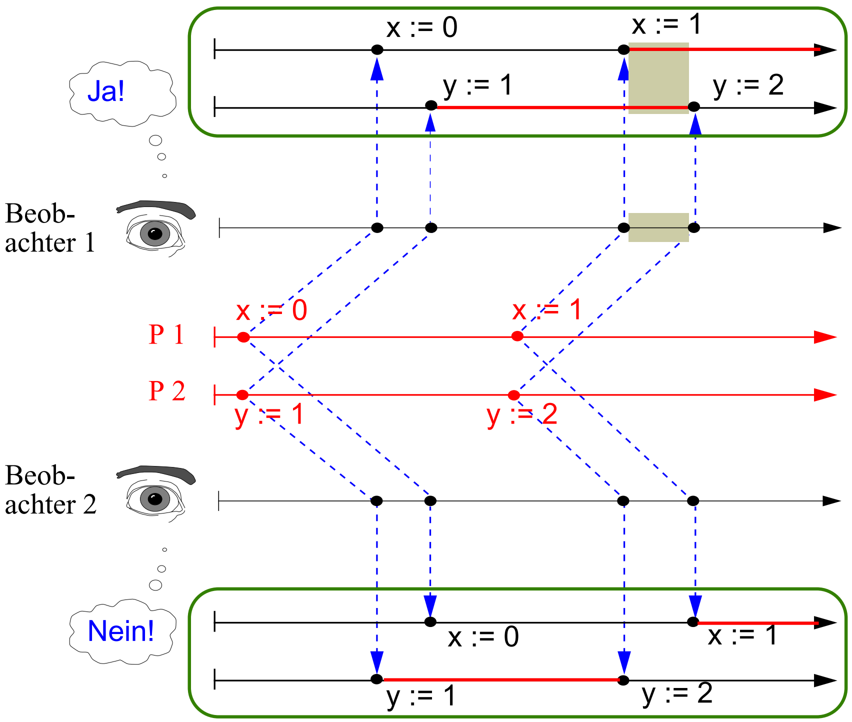


- Meldungen alle gleich schnell
- Beide Beobachtungen sind gleich “richtig”
- Die Beobachter stimmen bzgl. x = y nicht überein!

Aber was denn nun: gilt x=y in dieser Berechnung oder nicht?

Das "Entdecken" globaler Prädikate

Frage: Gilt in dieser Berechnung $\Phi \equiv (x = y)$?



• Meldungen sind alle gleich schnell

• Beide (kausaltreuen!) Beobachtungen sind gleich “richtig”

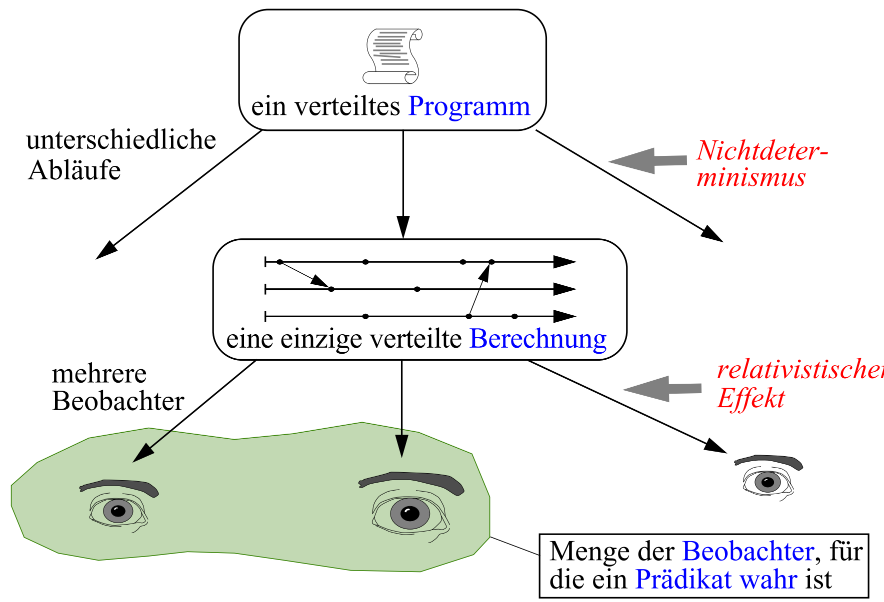
• Die Beobachter stimmen bzgl. x = y nicht überein!

Aber was denn nun: gilt x=y in dieser Berechnung oder nicht?

“Possible Worlds”

- Verschiedene Beobachter sehen *verschiedene Wirklichkeiten*

--> Frage, ob ein bestimmtes Prädikat gilt, ist u.U. sinnlos!



Konsequenz:

Es ist naiv (d.h. falsch!), einen verteilten Debugger zu entwickeln, mit dem man solche (im sequentiellen Fall “richtigen”) Fragen beantworten kann!

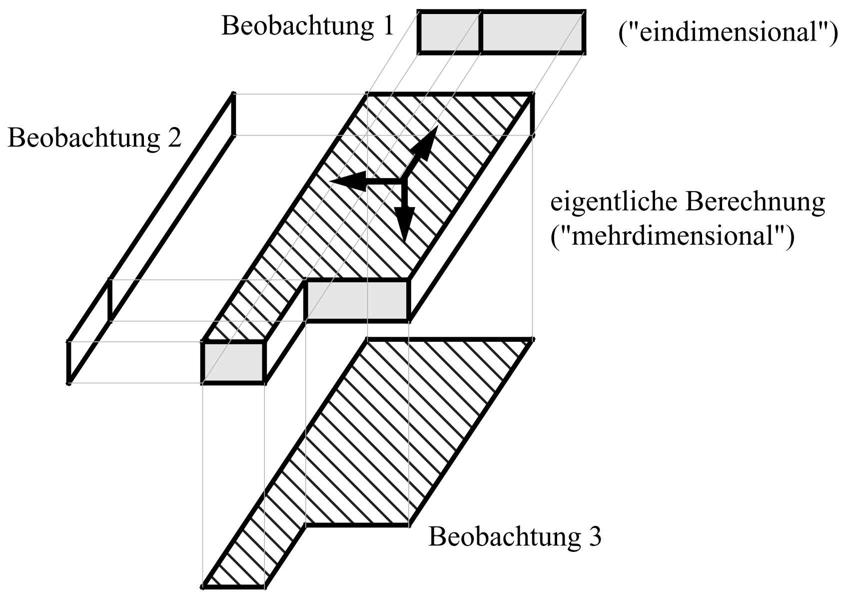
• im sequentiellen Fall: Berechnung = Beobachtung

• im verteilten Fall aber: Berechnung ≠ Beobachtung

• Gültigkeit von Prädikaten ist eine Eigenschaft einer Beobachtung, nicht einer Berechnung!

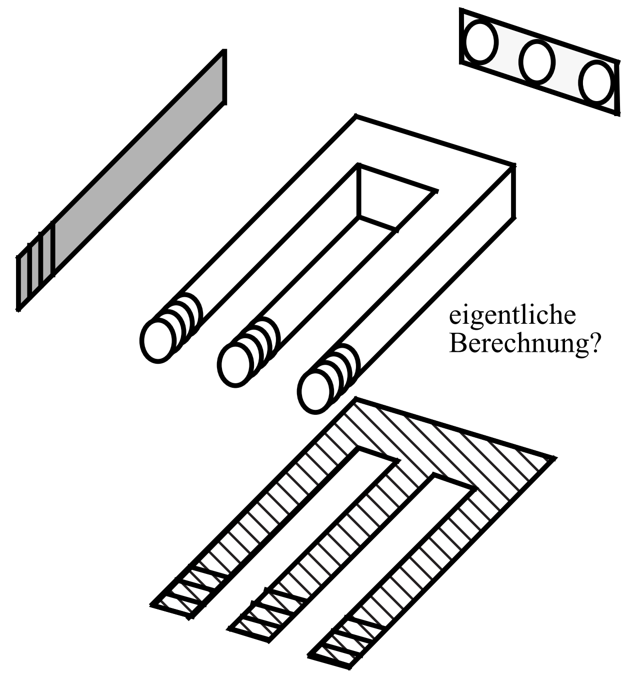
• gibt es sinnvolle beobachterinvariante Prädikate?

Beobachtung, Bild und Wirklichkeit



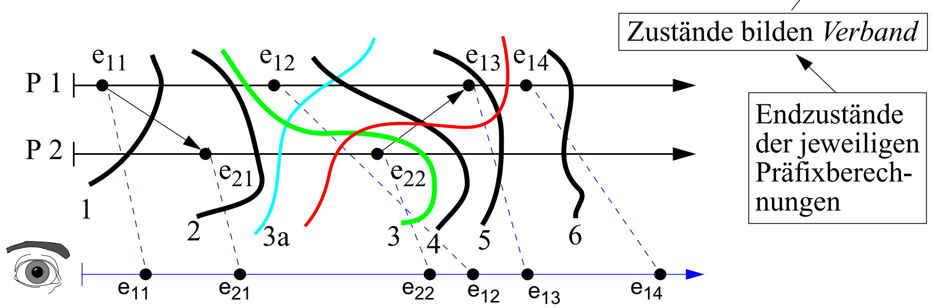
- Einige Eigenschaften gehen durch die Abbildung verloren
  - hier: kausale Unabhängigkeit
- Essentielle Eigenschaften sollen erhalten bleiben
  - hier: kausale Abhängigkeit
- Lässt sich die eigentliche Berechnung aus allen Beobachtungen rekonstruieren?
  - ja: Schnitt aller Linearisierungen ist Halbordnung selbst

Wirklichkeit ?

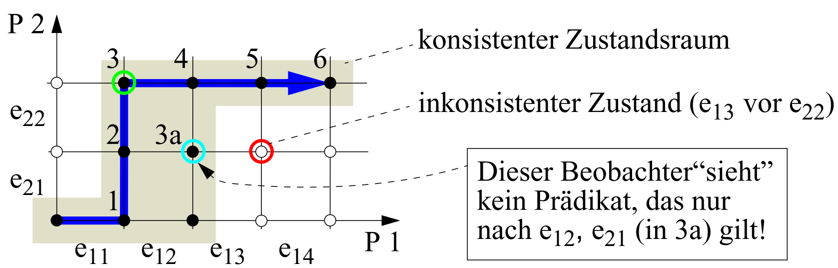


- Sind die verschiedenen Beobachtungen “verträglich”?
- Die beobachtete Wirklichkeit kann weitaus merkwürdiger sein, als eine Beobachtung vermuten lässt!
- Lassen sich die verschiedenen Beobachtungen zu einem konsistenten Bild zusammenfügen?

Das n-dimensionale Zustandsgitter



- *Beobachtung* =
  - Lineare Folge von Ereignissen
  - Folge damit assoziierter globaler Zustände



- *Beobachtung* = Pfad von links unten nach rechts oben
  (*kausaltreue Beobachtung,* wenn dieser sich nur im "erlaubten" Bereich aufhält)

- Ein Beobachter sieht alle *Ereignisse*, aber nur eine Teilmenge aller möglichen *Zustände*!
  --> Beobachter kann Prädikate übersehen!
- Mit einem Schnappschussalgorithmus kann nur eine *einzige* (i.a. *lückenhafte*) von vielen Beobachtungen erlangt werden

Fragen zum Beobachtungsbegriff

1) Gibt es zu jedem kons. Zustand s einer Berechnung (mind.) einen Beobachter, der diesen "wahrnimmt"?

2) Seien s, s' zwei bel. (kons.) Zustände. Gibt es immer eine Beobachtung Obs mit $s \in$ Obs und $s' \in$ Obs?

3) Können zwei konkurrente Zustände s, s' (d.h. weder $s<s'$ noch $s'<s$) in der selben Beobachtung auftauchen?
   (Ordnungsrelation entspr. Verband bzw. Präfixberechnung)

4) Es sei $s \in$ Obs und s später als s'. Ist dann $s' \in$ Obs?

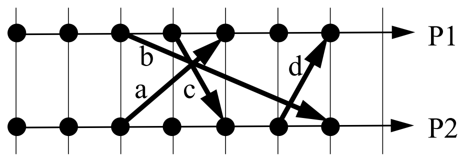
5) Sei $s \notin$ Obs. Gibt es stets ein $s' \in$ Obs, welches später als s ist? ... früher als s ist?

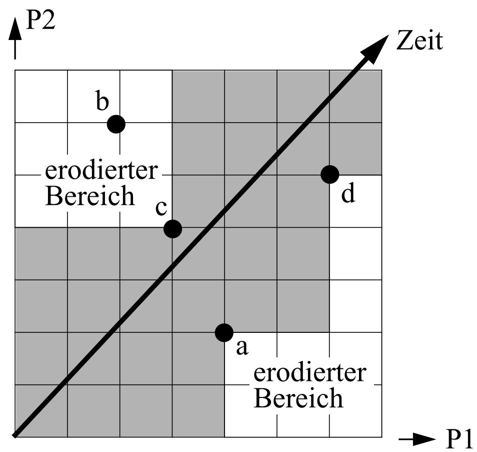
6) In einer Beobachtung Obs = $s_1,...,s_k$ sei $s_i \in$ Obs und $e \in s_i$ (Zustand aufgefasst als Menge der bis dahin vergangenen Ereignisse)
   a) Gilt dann $e \in sj$ für alle $sj \in$ Obs mit $j > i$? Interpretation?
   b) Es sei $e' < e$. Gilt dann $e' \in si$?

7) Sind alle Beobachtungen der selben (endlichen) Berechnung gleich lang?

Der erodierte Zustandshyperwürfel

- Hier: zwei Prozesse --> 2-dimensionaler Würfel



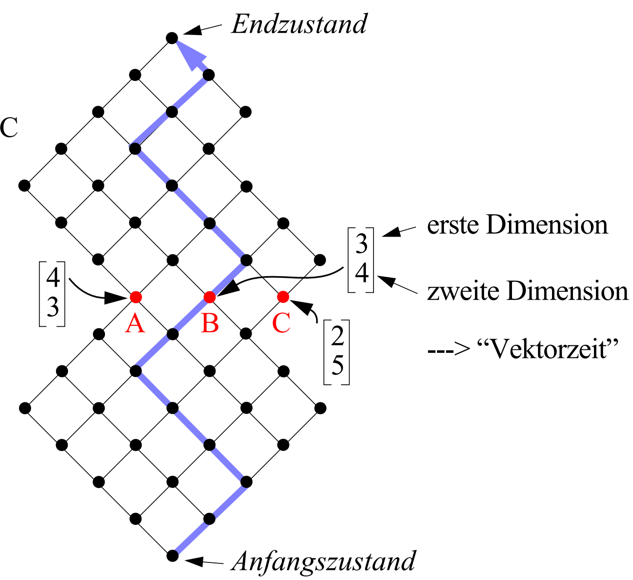


- "Erosion" der inkonsistenten Zuständen von den Ecken her
    - keine Nachricht wird empfangen, bevor sie gesendet wurde
    - ein Prozess blockiert in einer Empfangsanweisung, bis eine Nachricht verfügbar ist (und das zugehörige send somit ausgeführt wurde)
    - Nachrichten zur Synchronisation: sorgen dafür, dass kein Prozess sich zu schnell entfernt (zwingen Prozesse in den "Schlauch" von links unten nach rechts oben)

Der Verband konsistenter Zustände

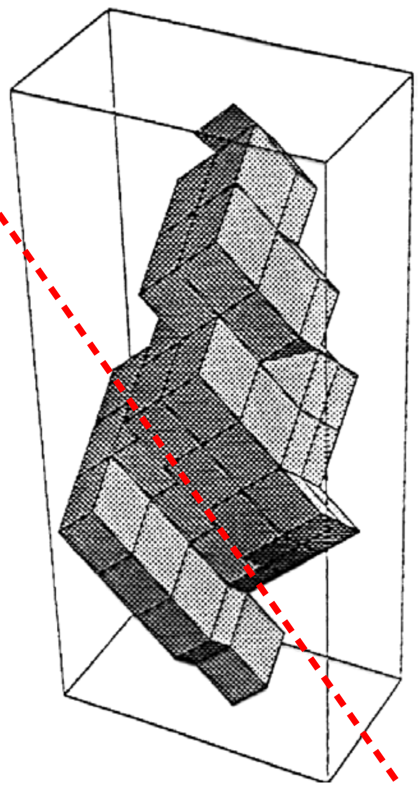
- Zu jeder Präfixberechnung gehört ein konsistenter Zustand.
- Die "echte" Folge globaler Zustände ist ein Pfad durch den Verband (ist aber unbekannt, wenn keine exakte globale Zeit existiert).

- drei "gleichzeitige" globale Zustände A, B, C
- Frage, durch welchen davon die Berechnung verlief, ist sinnlos!
- Äquivalenzklasse [A, B, C] (alle Zustände mit sieben Ereignissen)
- man weiss nur, dass die Berechnung durch diese *Klasse* lief



- Konsistente Zustände bilden einen *Verband*
    - früherer, späterer globaler Zustand; abgeschlossen bzgl. "sup" und "inf"
    - visualisiert als eine kompakte Menge (ohne "Löcher")
    - Unterverband des Verbandes *aller* globalen Zustände

Eine 3-fach verteilte Berechnung



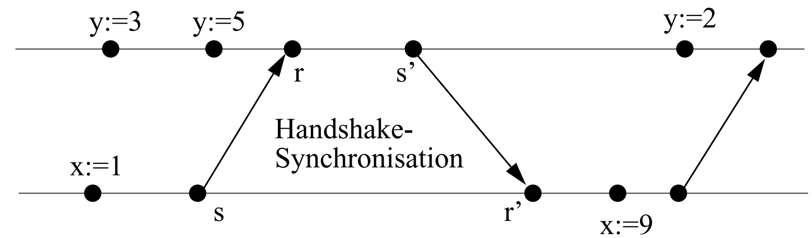
[Claude Jard et al., Rennes, Frankreich]

- 3-dimensionales Gebilde (als "kompakte Menge")
- Synchronisation --> "Kante" oder "Kerbe" auf der Oberfläche
- “Engpässe” werden so sichtbar!

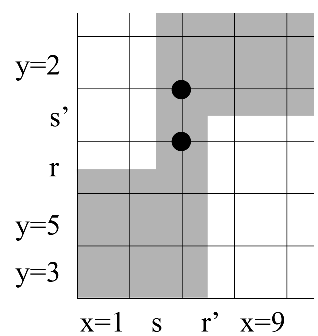
Kann man auch noch Berechnungen mit mehr Prozessen so visualisieren?

Synchronisationsengpässe

- Hier sieht *jeder* kausaltreue Beobachter "kurzzeitig" y=5, x=1:



- Der Zustand y=5, x=1 wird also auf alle Fälle angenommen; er ist daher "unvermeidlich"
- Verallgemeinerung: "Barrier Synchronisation"
  (erst wenn alle Prozesse die Barriere erreicht haben, dürfen sie weiterlaufen)

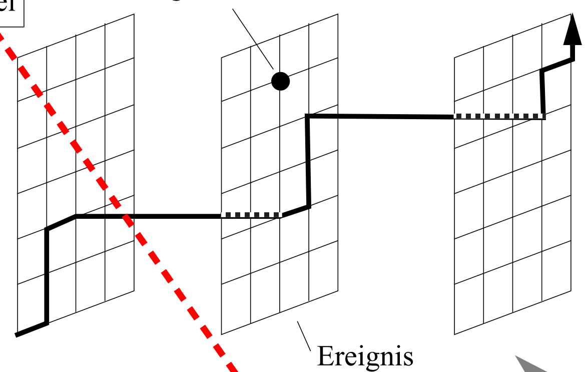


Durch die beiden markierten Zustände muss jeder Beobachter; dort gilt jedoch x=1, y=5

==> Mass für die Parallelität bzw. Unabhängigkeit?
(ein gutes numerisches Mass für "Parallelität" zu finden ist schwierig!)

Dualismus der Diagramme

Erodierter Hyperwürfel



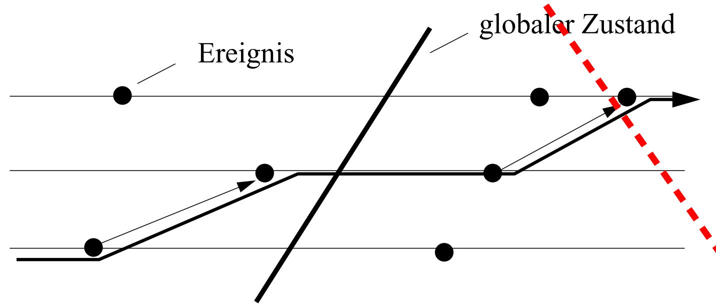
Hyper-ebene

Punkt --> globaler Zustand
Schnitt(ebene) --> Ereignis
Pfad --> Kette von Zuständen

Beide Diagramme repräsentieren die gleiche Berechnung

Punkt --> Ereignis
Schnitt --> globaler Zustand
Pfad --> Kette von Ereignissen

Zeitdia-gramm



Konsequenzen

- Prädikate gelten i.a. nur relativ zu Beobachtern, nicht für eine Berechnung als solche!
- Debugging: Nächster Schritt / Zustand nicht eindeutig
- Debugging: “stop when <condition>” i.a. sinnlos
- Polynomielle Zahl von Zuständen
- Exponentielle Zahl von Beobachtungen

--> i.a. ziemlich hoffnungslos!

- Ein einzelner Beobachter kann einen Zustand, in dem ein Prädikat gilt, "verpassen"

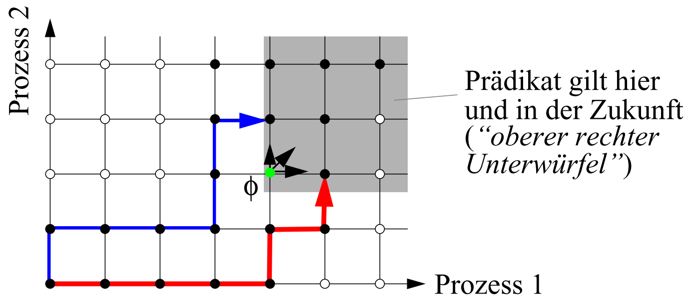
Genauere Spezifikation von Gültigkeit notwendig, z.B.:

- Prädikat Φ ist *aus Sicht von Beobachter X* erfüllt
- Prädikat Φ wird *von wenigstens einem* Beobachter wahrgenommen
- Prädikat Φ wird *von allen* Beobachtern wahrgenommen

Stabile Prädikate

- Informell: Monoton - "einmal wahr, immer wahr"
- Def.: wenn c1 < c2, dann Φ(c1) ==> Φ(c2)

Halbordnung im Zustandsverband



- Sind *beobachterunabhängig* (possibly Φ = definitely Φ)!
  - jeder Beobachter muss durch den oberen rechten Würfel (Fairness...)
  - lassen sich daher einfach mit einer einzigen Beobachtung feststellen (jede andere Beobachtung wird Φ früher oder später ebenfalls entdecken)
- Für zwei Beobachtungen B_1, B_2 gilt: Falls B_1 Φ "entdeckt", dann gibt es einen *gemeinsamen späteren Zustand* (Verbandseigenschaft!) von B_1, B_2, bei dem Φ gilt
  - spätestens der Endzustand (bei endlichen Berechnungen)
  - B_1 kann z.B. die "echte" Ereignisfolge in Realzeit sein, B_2 eine Beobachtung
- Ein *gelegentlicher* (konsistenter) Schnappschuss genügt!
  - wenn der Schnappschussalgorithmus die Gültigkeit von Φ ermittelt, dann gilt Φ "jetzt" tatsächlich
  - wenn Φ "jetzt" gilt, dann meldet dies ein (jetzt gestarteter) Schnappschussalg.
- Es gibt wichtige stabile Prädikate, z.B. Terminierung, Garbage, Deadlock...
- Aber woher weiss man eigentlich, ob bzw. dass ein Prädikat stabil ist?

Logische Zeit in verteilten Systemen

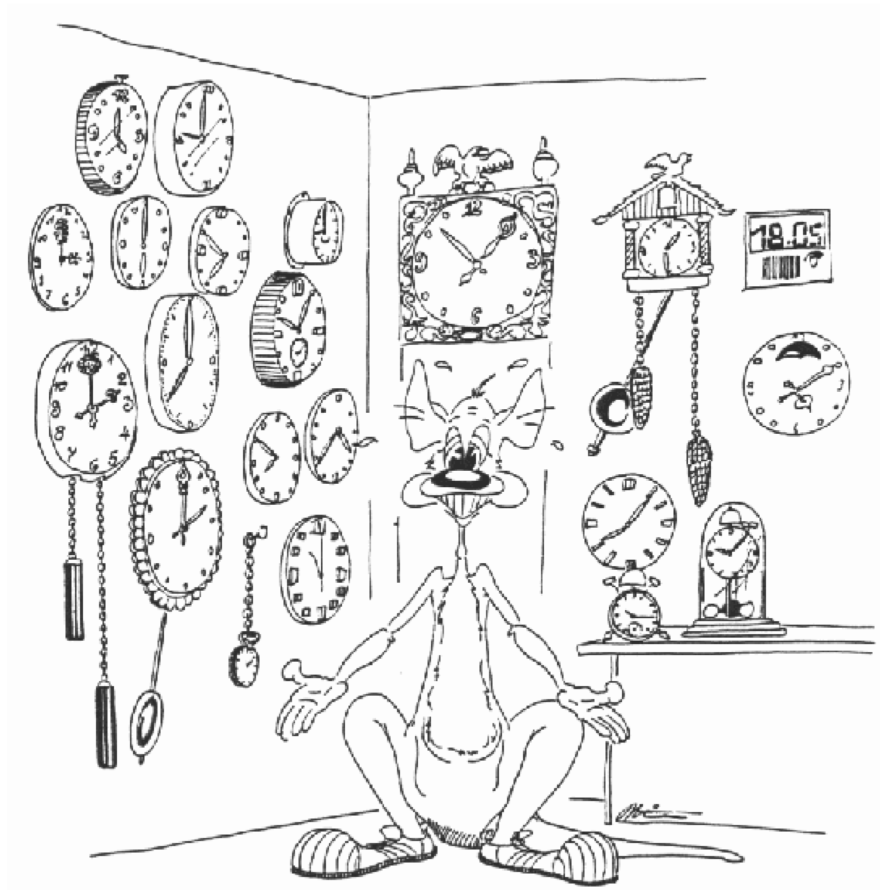
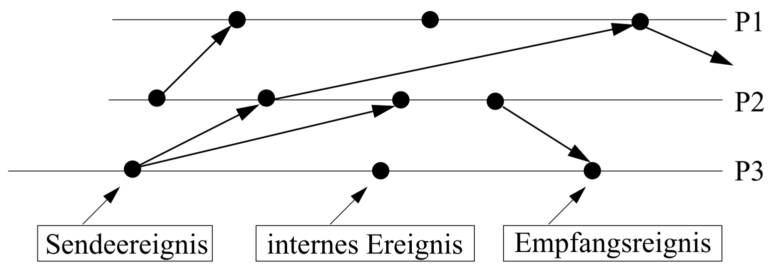


Bild: R. G. Herrtwich, G. Hommel

Kausalrelation ("Happened Before", Lamport 1978)

- *Zur Wiederholung:*



- interessant: von links nach rechts verlaufende "Kausalitätspfade" (bestehend aus Nachrichtenpfeilen + Teilstücken auf Prozessachsen)

- *Kausalrelation* ‘<‘ auf der Menge E aller Ereignisse:

> “Kleinste” transitive Relation auf E, mit **x** < **y** wenn:
>
> 1) **x** und **y** auf dem gleichen Prozess stattfinden und **x** vor **y** kommt, *oder*
>
> 2) **x** ist ein <u>Sendeereignis</u> und **y** ist das korrespondierende <u>Empfangsereignis</u>

- In einem Zeitdiagramm gilt für je zwei Ereignisse e, e’ die Relation e<e’ genau dann, wenn es einen Kausalitätspfad von e nach e’ gibt

Logische Zeitstempel von Ereignissen

- Verteilte Berechnung abstrakt: n Prozesse, halbgeordnete Ereignismenge E, Nachrichten (Sende- / Empfangsereignis)

- Zweck: Ereignissen eine Zeit geben ("dazwischen" egal)

- Gesucht: Abbildung C: E --> H

Clock

"Zeitbereich": Halbgeordnete Menge --> "füher", "später"

- Für e ∈ E heisst C(e) *Zeitstempel* von e

- C(e) bzw. e *früher* als C(e’) bzw. e’, wenn C(e) < C(e’)

- Wie soll H aussehen?

z.B.:

- **N** (lineare Ordnung)
- **R** (bzw. REAL-Datentyp)
- Potenzmenge von E
- $\mathbf{N}^n$ (d.h. n-dim. Vektoren)

- Sinnvolle Forderung:

Kausalrelation ("Pfad im Diagramm")

Uhrenbedingung: e < e’ ==> C(e) < C(e’)

Ordnungshomomorphismus

Zeitrelation “früher”

Interpretation ("Zeit ist kausaltreu"):

Wenn ein Ereignis e ein anderes Ereignis e’ beeinflussen kann, dann muss e einen kleineren Zeitstempel als e’ haben

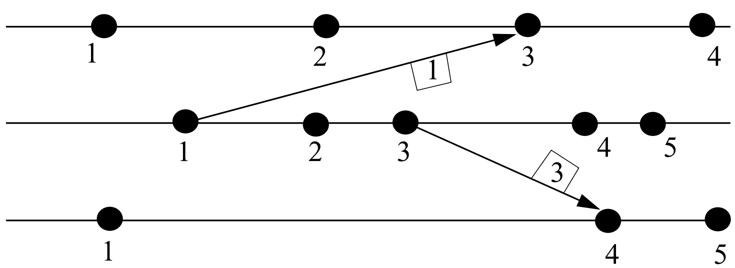
Logische Uhren von Lamport

Commun. ACM 1978:
Time, Clocks, and the Ordering of Events in a Distributed System

$C: (E,<) \dashrightarrow (\mathbf{R},<)$ Zuordnung von Zeitstempeln

Kausal-relation (oder **N**)

$e < e' \implies C(e) < C(e')$ Uhrenbedingung



Protokoll zur Implementierung der Uhrenbedingung:

- Lokale Uhr (= Zähler) *tickt* "bei" *jedem* Ereignis
- Sendeereignis: Uhrwert mitsenden (*Zeitstempel*)
- Empfangsereignis: *max*(lokale Uhr, Zeitstempel)

zuerst! danach "ticken"

Behauptung:
Protokoll respektiert Uhrenbedingung

Beweis: Kausalitätspfade sind monoton...

Kausale Unabhängigkeit

- *Def*. ["kausal unabhängig"]: $a \| b \Leftrightarrow: \neg(a < b) \wedge \neg(a > b)$

Beachte: $\|$ ist *nicht transitiv*! (Gilt eigentlich $a\|a$?)

- *Behauptung*: $C(a) = C(b) \implies a\|b$

Bew.:

$C(a) = C(b) \implies \neg(C(a) < C(b)) \wedge \neg(C(a) > C(b))$

$\implies \neg(a < b) \wedge \neg(a > b)$

Verwende $\neg(C(x) < C(y)) \implies \neg(x < y)$ aus der Uhrenbedingung!

$\implies a\|b$ (Def.)

- Gilt die Umkehrung der Uhrenbedingung?

Nein, $C(e) < C(e') \implies e < e'$ gilt nicht!

Es gilt nur: $C(e) < C(e') \implies e < e'$ *oder* $e\|e'$

vgl. Beispiel

Das heisst:

Aus den Zeitstempeln lässt sich nicht (immer) schliessen, ob zwei Ereignisse kausal voneinander abhängig sind, oder ob sie unabhängig sind!

==> Aber wozu sind solche Zeitstempel dann gut?

Vektorzeit: Motivation

Umkehrung der Uhrenbedingung gilt nicht für Lamport-Zeit

- $C(e) < C(e') \implies e < e'$ gilt nicht!
- es gilt nur: $C(e) < C(e') \implies e < e'$ oder $e \| e'$

Zeit := vergangene Zeit

:= Vergangenheit

:= Menge vergangener Ereignisse

viele Uhren messen die Zeit, indem sie *vergangene* Sekunden zählen

vgl. dies mit der *Lamport-Zeit* ("lokal vergangen")

Kausalrelation

Zeit(e) := $\{e' \mid e' \leq e\}$ = *Kegel* von e

Genauer: Zeitstempel eines Ereignisses

Kann durch lokal späteste Ereignisse repräsentiert werden (linksabgeschlossen)

Hiervon gibt es n Stück (n = Anzahl der Prozesse)

!

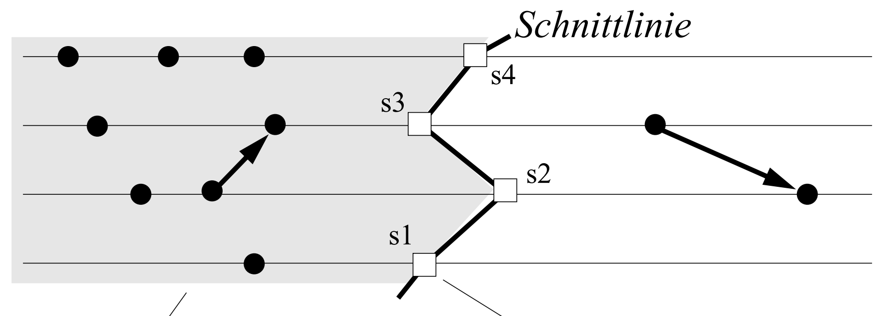
--> *Zeitstempel* ist n-dimensionaler Vektor

--> *Zeit* ist Menge n-dimensionaler Vektoren

--> *Uhr* ist ein array C[1:n]

zum Anzeigen von Zeitvektoren

Schnitt und Schnittlinie



Schnittereignis
(zusätzlich zu "Basisereignissen" je ein Ereignis pro Prozess "festlegen")

Vergangenheit V
$\{e \mid \exists s_i: e <_l s_i\}$

- Schnittlinie trennt Zeitdiagramm / Ereignismenge in zwei disjunkte Mengen "Vergangenheit" / "Zukunft"

- Bemerkung: $e \in V \wedge e' <_l e \implies e' \in V$
  (linksabgeschlossen bzgl. lokaler Kausalrelation)

Denkübung: Man vergleiche den Begriff des *Schnittes* (insbes. des *konsistenten Schnittes*, vgl. nächste Seite) mit dem früher erwähnten Begriff der *Präfixberechnung*! Man beachte auch die Halbordnungsstruktur bzw. Verbandsstruktur dieser Begriffe.

Konsistente Schnitte

Def. **Schnitt**:

$S \subseteq E$ heisst *Schnitt* von E, falls $e \in S \wedge e' <_1 e \Longrightarrow e' \in S$

(d.h. Schnitt wird mit seiner Vergangenheit identifiziert)

Def. **konsistenter Schnitt**:

$S \subseteq E$ heisst *konsistent*, falls $e \in S \wedge e' < e \Longrightarrow e' \in S$

Def.: Schnitt S *später* als S' :<==> $S' \subseteq S$

bzw. $\subset$ bei "strikt später"

Schreibweise auch: $S' < S$

Beh.: Jeder konsistente Schnitt ist ein Schnitt

Bew.: $<l \subseteq <$

Bem.: Schnitt(linie) inkonsistent <==>
$\exists$ "Nachricht aus der Zukunft"

Bem.: Schnitt(linie) konsistent <==>
Schnittereignisse paarweise kausal unabhängig

Bew. als Übung

Bem.: Schnitt(linie) konsistent <==>
lässt sich senkrecht darstellen (Gummibandtransf.)

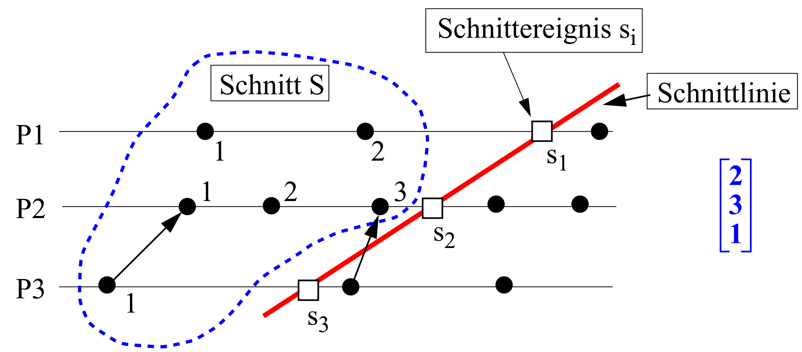
Bew. bereits bekannt: Diagramm auseinanderschneiden und versetzen

Vektorzeitstempel von Schnitten

Zeitstempel $\tau(S)$ eines Schnittes S ist ein Vektor aus $\mathbf{N}^n$

alle Ereignisse, die links von einer Schnittlinie liegen

Anzahl der Prozesse



$\begin{bmatrix}2\\3\\1\end{bmatrix}$

lokal!

Def. $\tau(S)[i] := |\{e \in E_i \mid e <_1 s_i\}| := |S \cap E_i|$

(für jede Komponente i mit $1 \leq i \leq n$)

Interpretation:

$\tau(S)[i]$ ist die "Stelle", wo die Prozessachse von P_i durch die Schnittlinie geschnitten wird

Beachte: man kann zu *konsistenten* und *inkonsistenten* Schnitten den zugehörigen Zeitvektor definieren!

Kausale Vergangenheit eines Ereignisses

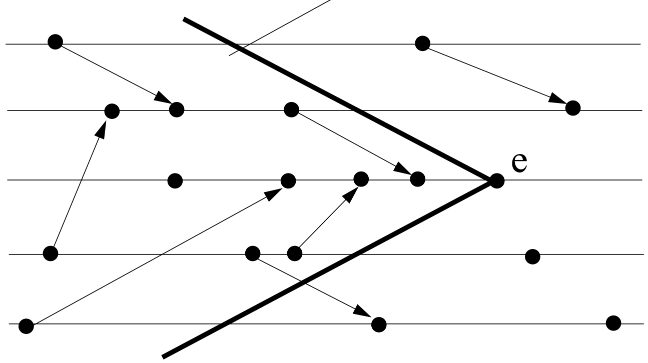
Def. *kausale Vergangenheit* $\downarrow(e)$ eines Ereignisses e:

$$\downarrow(e) = \{e' \mid e' \leq e\}$$

Beh.: $\downarrow(e)$ ist ein konsistenter Schnitt

Bew. als Übung

Der von e *erzeugte* Schnitt
("Kegel" / "causal history")



Beh.: $e' \leq e \iff e' \in \downarrow(e)$

Beh.: $e \parallel e' \iff \neg(e \in \downarrow(e')) \wedge \neg(e' \in \downarrow(e))$

(Bew. klar)

Vektorzeitstempel von Ereignissen

Kausale Vergangenheit

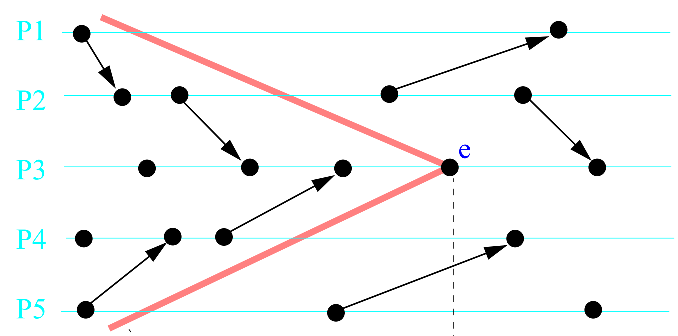
- Bezeichne $\downarrow e$ den *Kegel* $\{e' \in E \mid e' \leq e\}$ von e
  - jeder Kegel ist ein konsistenter Schnitt (da linksabgeschlossen) ("Kegelmantel" könnte also als senkrechte Linie gezeichnet werden!)
  - Repräsentation durch die n lokal am weitesten rechts liegenden Ereignisse

Menge der Ereignisse von P_i

- Dann definiere $\tau(e) := \tau(\downarrow e)$

  Also: $\tau(e)[i] := |\{e' \in E \mid e' \leq e\} \cap E_i| := |\{e' \in E_i \mid e' \leq e\}|$

  - das heisst: Zeitstempel eines Ereignisses = Zeitstempel seines Kegels

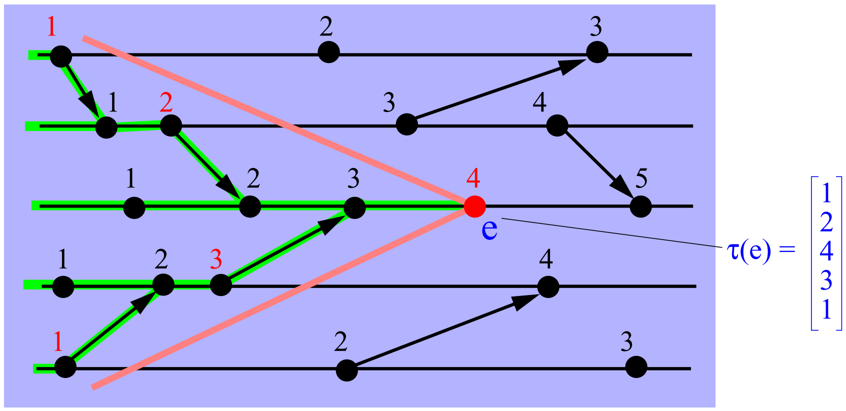


Schnittlinie in der Form eines Kegels (vgl. "Lichtkegel" der relativistischen Physik): $\downarrow e$ ist ein konsistenter Schnitt

$$\begin{bmatrix}1\\2\\4\\3\\1\end{bmatrix}$$ $\tau(e)$: Vektorzeitstempel von e

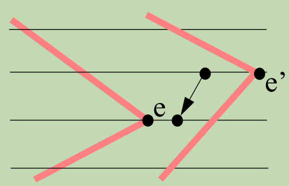
τ(e) < τ(e')

- Jeder Prozess numeriert seine Ereignisse lokal durch



- Vektor τ(e) repräsentiert gesamte kausale Vergangenheit des Ereignisses e
  - Veranschaulichung durch einen "Kegel" {e' | e' ≤ e}
  - Nummer des jeweils lokal letzten kausal vorangehenden Ereignisses steht in der jeweiligen Komponente des Vektors

- Interpretation von τ(e) < τ(e'):
  - e liegt in der kausalen Vergangenheit von e'
  - Kegel von e ist ganz im Kegel von e' enthalten



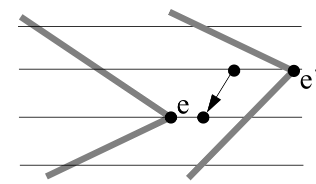
Def. "Zeitstempelarithmetik"

$$\begin{bmatrix}1\\3\\4\\3\\2\end{bmatrix} \leq \begin{bmatrix}1\\7\\4\\6\\2\end{bmatrix} \qquad \begin{bmatrix}1\\3\\4\\3\\7\end{bmatrix} \parallel \begin{bmatrix}5\\3\\8\\3\\2\end{bmatrix}$$

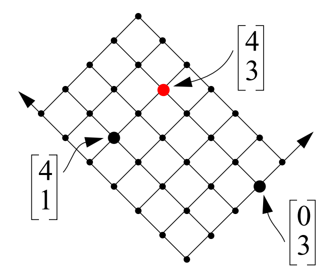
vergleichbar (komponentenweise ≤) "konkurrent"

'<' definiert als "≤ aber ≠"

Interpretation von τ(e) < τ(e'):
- e liegt in der kausalen Vergangenheit von e'
- Kegel von e ist im Kegel von e' enthalten



$$\sup\left(\begin{bmatrix}1\\4\\2\\3\\7\end{bmatrix}, \begin{bmatrix}8\\3\\4\\3\\2\end{bmatrix}\right) = \begin{bmatrix}8\\4\\4\\3\\7\end{bmatrix}$$



sup = komponentweises Maximum

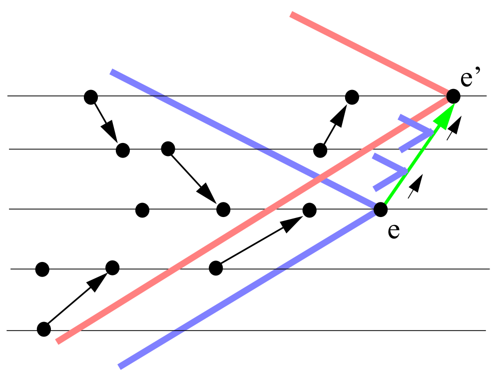
Implementierung der Vektorzeit

- Idee: Analog zur Lamport-Zeit
  (hier allerdings stets vektoriell!)

- Nachrichten enthalten die gesamte kausale Vergangenheit des Senders ==> Zeitvektor des Sendeereignisses

- Bei Empfang einer Nachricht:
  - Vereinigung der Kegel
  - ==> Supremum der Vektoren

Wissen über vergangene Ereignisse vereinigen



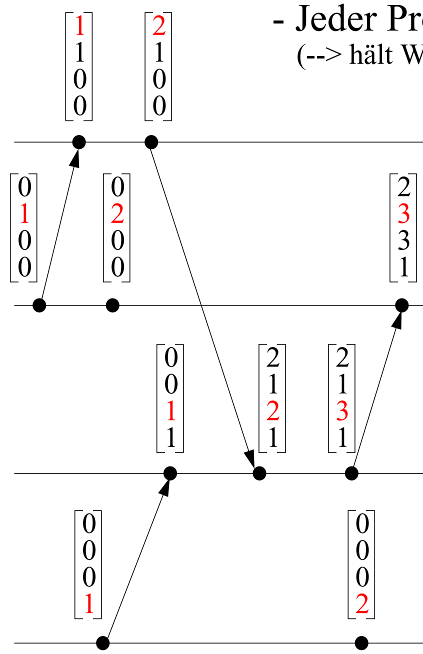
Mitschleppen des Kegels des Sende-ereignisses und Vereinigung mit dem Kegel des Empfangsereignisses

--> "induktiv": ein Ereignis hat ein "vollständiges Wissen" über alle seine vergangenen Ereignisse

Propagieren des Zeitwissens

(--> Implementation der Vektorzeit)

Andere Zeiten, andere Sitten



- Jeder Prozess besitzt eine *Vektoruhr*
  (--> hält Wissen über vergangene Ereignisse)

- *bei jedem Ereignis:*
  eigene Komponente erhöhen
- *beim Senden:*
  neuen Vektor mitsenden
- *beim Empfangen:*
  komponentenweises Maximum der beiden Vektoren

Vereinigung der beiden Kegel

Kausalrelation

bzgl. Zeitvektor

- *Behauptung*: $e < e' \Leftrightarrow \tau(e) < \tau(e')$

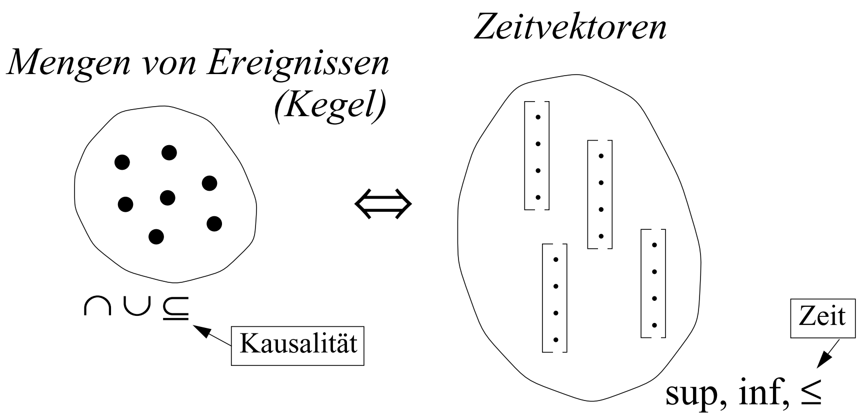
- *Anschauliche Interpretation*:
  - $\tau(e) \leq \tau(e') \Leftrightarrow$ es gibt eine Kausalkette von e zu e'

monoton bzgl. Zeitvektoren!

- *Korollar*: $e \parallel e' \Leftrightarrow \tau(e) \parallel \tau(e')$

Interpretation: Genau die "gleichzeitigen" Ereignisse beeinflussen sich nicht ggs.

Rechnen mit Ereignismengen



Mengentheoretische Operationen $\Leftrightarrow$ Algebraische Operationen (--> “rechnen”)

Verbandsstruktur auf 2^E (Ideale) $\Leftrightarrow$ Produduktverband auf $\mathbf{N}^n$

Ordnungstheoretische Eigenschaften $\Leftrightarrow$ Algebraische Eigenschaften

---

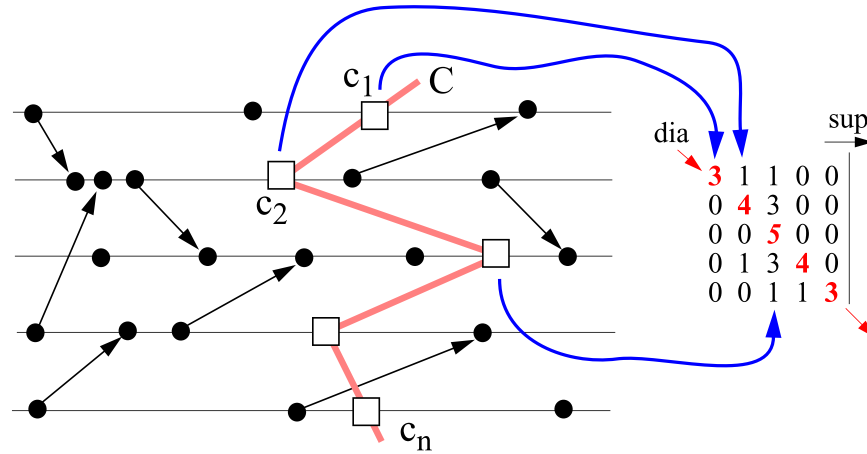
Vektorzeit und Vektoruhren ermöglichen eine "operationale Manipulation" der Kausalrelation

Schnittmatrix

- Schnittmatrix \$ eines Schnittes C (mit Schnittereign. c_i):

$$\$(C) := (\tau(c_1), \tau(c_2),...,\tau(c_n))$$

d.h. Schnittereignisvektoren c_i als Spaltenvektoren



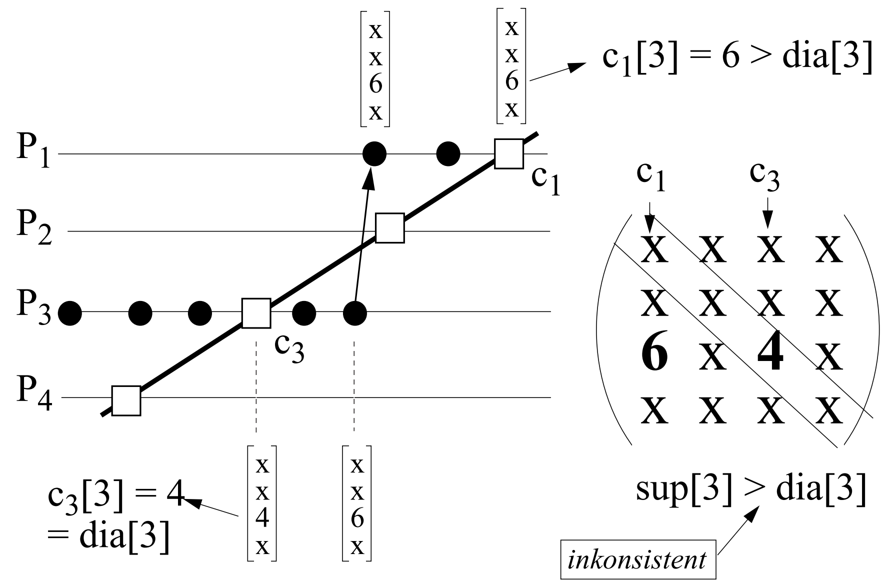
Frage: Kann man an den Schnittmatrizen etwas über die Schnitte erkennen? (z.B.: ob später, früher; ob konsistent...)

$\boxed{\text{C konsistent} \Leftrightarrow \text{dia}(\$) = \text{sup}(\$)}$

Diagonalvektor Zeilenmaximum

(d.h. Maximum einer Zeile ist das Diagonalelement)

Das "sup = dia"-Konsistenzkriterium



Ein Prozess (P_1) verschieden von P_3 weiss (bei c_1) etwas über lokale Ereignisse auf P_3, von denen P_3 selbst noch nichts weiss (d.h. die *nach* c_3 geschehen)

<==>

Es gibt einen Pfad von einem Ereignis auf P_3 *nach* c_3 zu einem Ereignis *vor* c_1

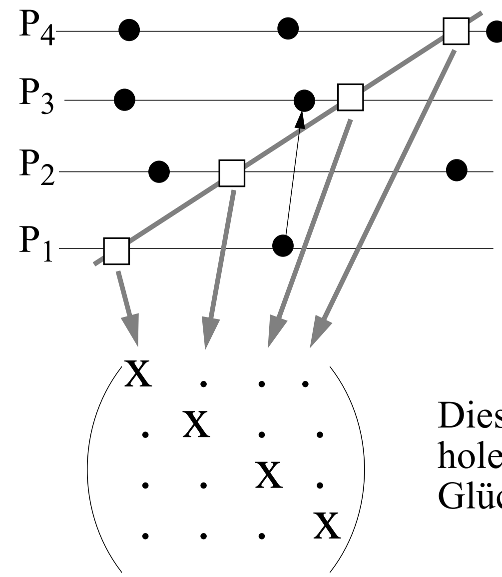
<==>

[Generalisierung über alle Indizes $i \neq j$]

Der Schnitt ist inkonsistent

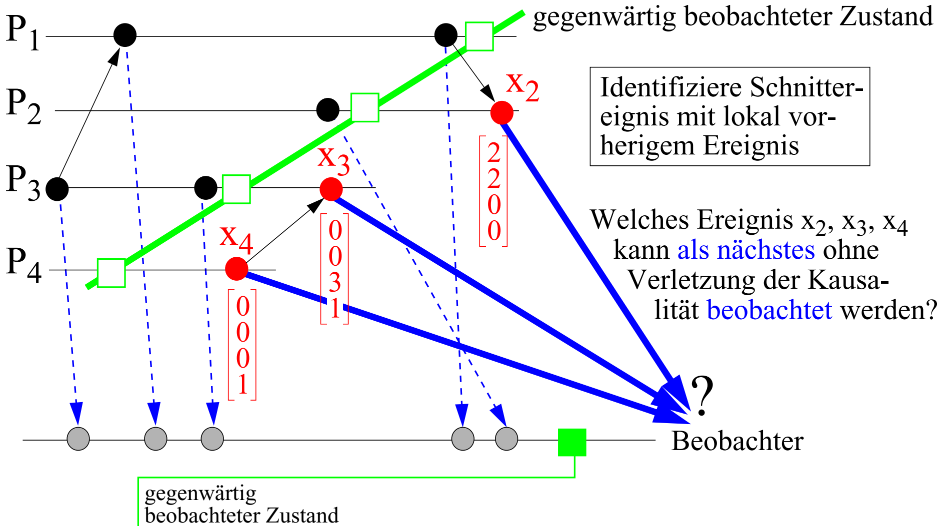
Implementierung konsistenter Schnappschüsse mit Vektorzeit?

Ein erster Ansatz: Alle Prozesse auffordern, ihren lokalen Zustand zu senden und testen, ob konsistent:



Dieses solange wiederholen, bis man einmal Glück hat...

Realisierung kausaltreuer Beobachter



$$\begin{pmatrix} \mathbf{2} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 1 & 0 & \mathbf{2} & 0 \\ 0 & 0 & 0 & \mathbf{0} \end{pmatrix}$$

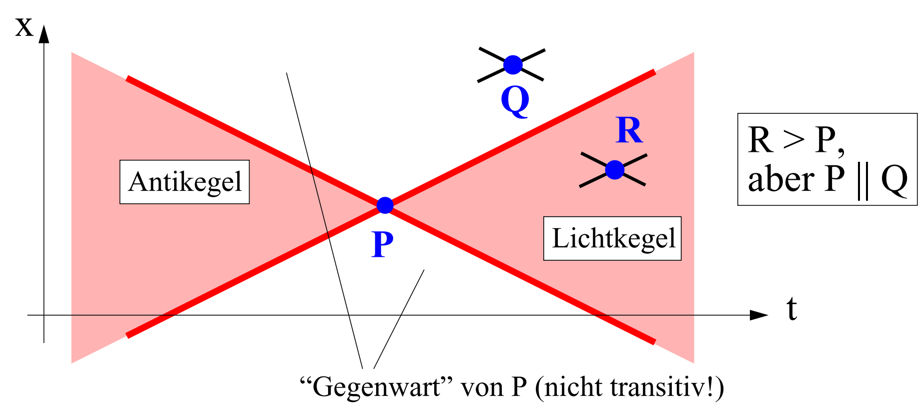
- Welche *Spalte* kann *ausgetauscht* werden? (x_2, x_4, aber nicht x_3)
- Beobachter merkt sich dia($); es muss *Zeitstempel ≤ dia($)* sein (ausgenommen die Diagonalkomponente)

- Strategie: dia($) = sup($) --> stets konsistent halten!
- Beobachter benötigt nur den Diagonalvektor, keine Matrix, um momentanen Zustand zu identifizieren

Prinzip: Verwende Vektorzeit um (indirektes) *Wissen* über "kausal frühere" Sendeereignisse zu kodieren:

"Dieses Ereignis hängt von einem anderen ab, das ich eigentlich erhalten haben müsste; also warte ich das andere erst ab..."

Vektorzeit und Minkowski-Raumzeit



Raumzeit	*Vektorzeit*
Halbordnung	Halbordnung
2-dimensionale Kegel bilden Verband (bzgl. Schnitt)	Zeitvektoren bilden Verband (sup)
Lorentz-Transformation lässt Lichtkegel invariant	Gummiband-Transformation lässt Kausalrelation invariant
Raumzeitkoordinaten ermöglichen Test, ob potentiell kausal abhängig: Mit u= (x_1, t_1), v= (x_2, t_2) prüfe $c^2(t_2-t_1)^2 - (x_2-x_1)^2 >= 0$	Zeitvektoren ermöglichen einfachen Test, ob potentiell kausal abhängig: (prüfe, ob in allen Komponenten kleiner)