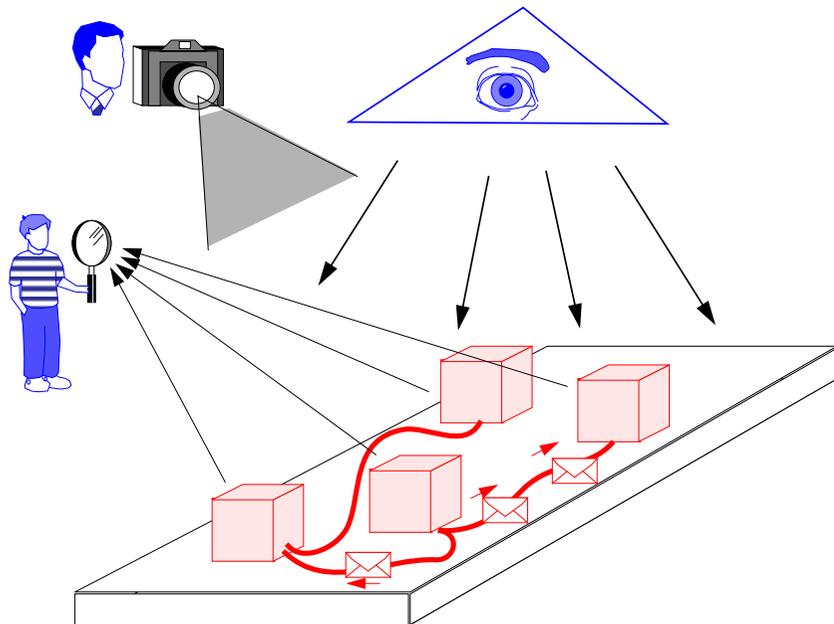


# Globale Zustände, Beobachtungen, Prädikate



## Kausal-konsistente globale Zustände

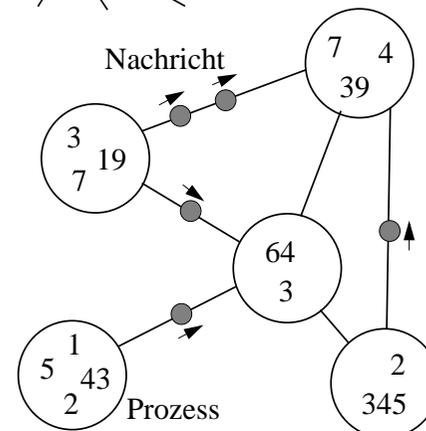
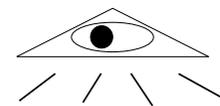
Webster:

*State* = a set of circumstances or attributes characterizing a person or thing at a given time

Gibt es "globale Zeit" in einem vert. System?

*Globaler Zustand* (zu einem Zeitpunkt):

Alle lokalen *Prozesszustände* + alle *Nachrichten*, die "unterwegs" sind



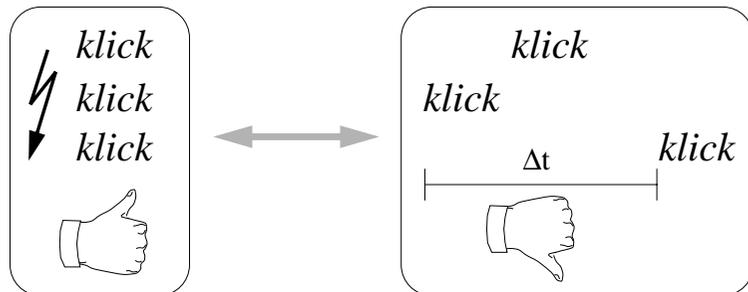
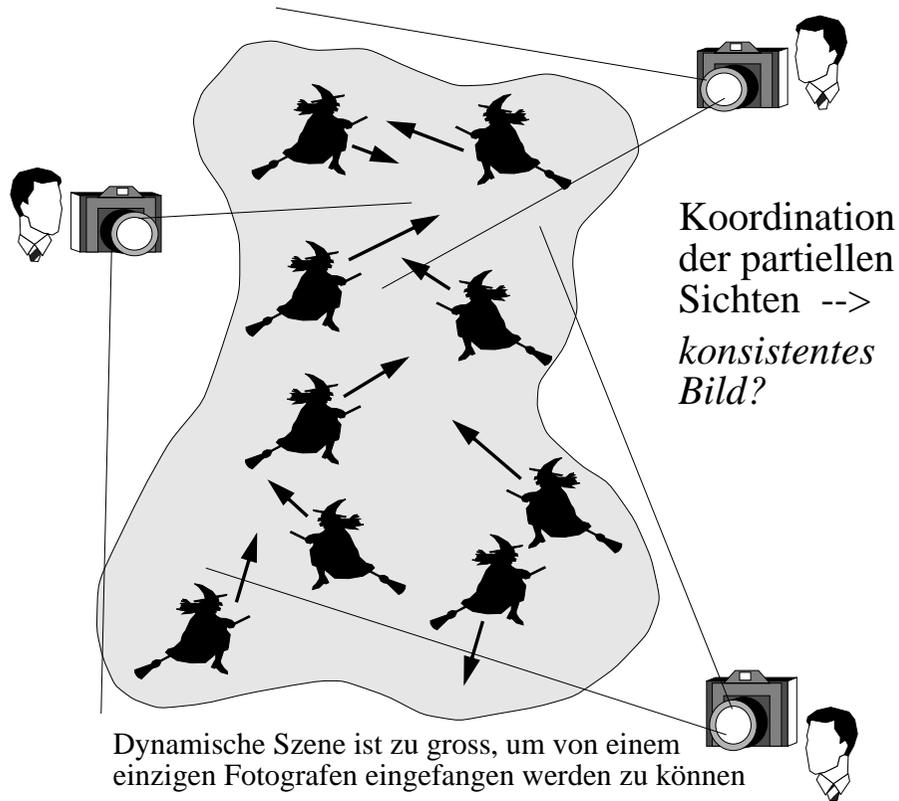
Schnappschuss

*Problem:*

Prozesszustände sind nur einzeln ("nacheinander") erfahrbar --> Konsistenz?

als ob alles gleichzeitig wahrgenommen wird

# Synchronisierte lokale Schnappschüsse?



# Das Schnappschussproblem

*Problem:* "Momentaner" Schnappschuss des globalen Zustands, ohne das System anzuhalten

*Realität:*

- *Volkszählung:* Stichzeitpunkt (geht hier nicht)
- *Inventur:* Einfrieren (unpraktisch)

*Anwendungen:*

- konsistenter Aufsetzpunkt für vert. Datenbanken
  - wie hoch ist die momentane Last?
  - Testen verteilter Systeme (gilt eine globale Eigenschaft?).
  - Deadlock: Existiert eine zykl. Wartebedingung?
  - ist die verteilte Berechnung terminiert?
  - ist ein bestimmtes Objekt "Garbage"?
  - ...
- } Prädikate über glob. Zuständen

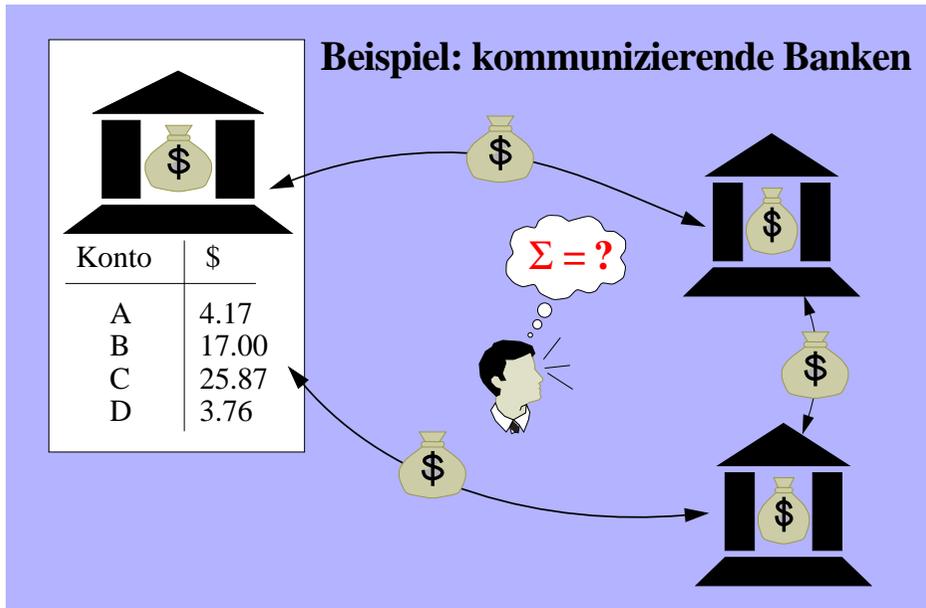
*Schwierigkeiten:*

- unmöglich, alle Prozesse gleichzeitig zu erwischen
- unbestimmte Nachrichtenlaufzeiten
- Nachrichten, die unterwegs sind, sieht man nicht
- ermittelter Zustand ist i.a. veraltet
- ... u.U. nie "wirklich" so gewesen
- ... u.U. inkonsistent

zumindest dies ausschliessen!

→ Schnappschussalgorithmus

# Beispiel: Kommunizierende Banken



## - Modellierung:

- **ständige Transfers** zwischen den Konten bzw. Banken
- **lokale atomare Aktionen**: alle Geldkonten *einer* Bank können "gleichzeitig" (also atomar und damit "lokal konsistent") untersucht werden

## - Wieviel Geld ist in Umlauf?

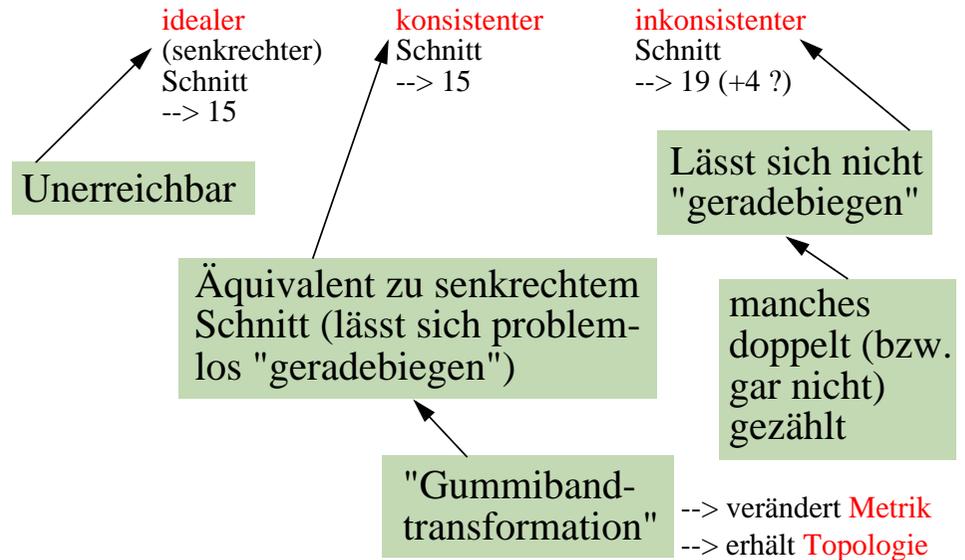
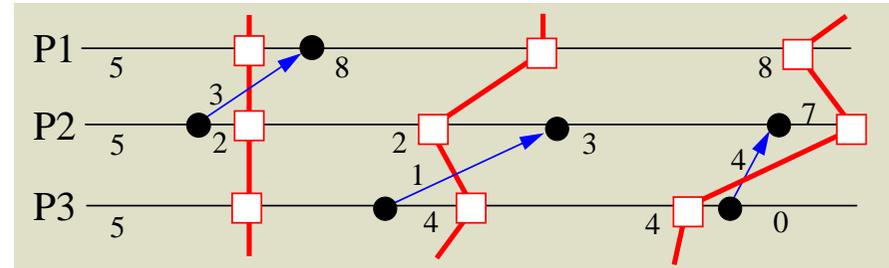
- **konstante** Geldmenge, oder
- **monotone** Inflation (--> untere Schranke für momentane Geldmenge)

## - Erschwerte Bedingungen:

- niemand hat eine **globale Sicht**
- gemeinsame **Zeit**?

# (In)konsistente Schnitte

Beispiel: Wieviel Geld ist in Umlauf?



Wie wir noch einsehen werden:

**Kausaltreues Beobachten  $\Leftrightarrow$  konsistente Schnitte**  
- wie erreicht man das?

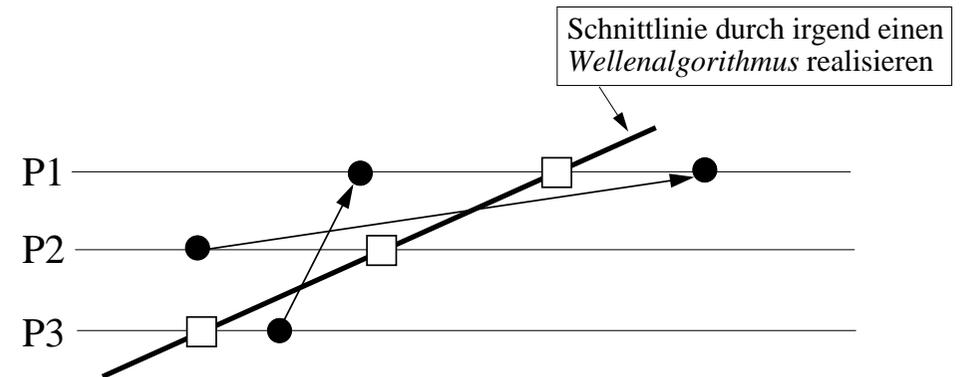
# Schnappschussalgorithmen: Zweck

- Liefern *konsistenten, möglichen, vergangenen* Zustand
- auch wenn der Zustand nur möglich gewesen wäre, aber gar nicht *wirklich* auftrat!
- aber was heisst schon "wirklich"?

- *konsistenter Zustand* = globaler Zustand entlang einer konsistenten Schnittlinie (keine Nachricht "aus der Zukunft")
- nur über konsistenten Zuständen können *globale Prädikate* sinnvoll bestimmt werden (da diese "äquivalent" zu Zuständen entlang senkrechter Schnittlinien sind); in diesem Sinne sind solche Algorithmen wichtig!
- ein Prädikat heisst *stabil* (oder *monoton*), wenn es nie wieder aufhört zu gelten, nachdem es gilt (also in allen zukünftigen Zuständen gilt); z.B. Terminierung, Garbage, Deadlock...

- Falls Prädikat stabil: "entdecken" dieses Prädikat (==> "stable property detection algorithm")
  - bei stabilen Prädikaten ist "möglich... vergangen" sogar brauchbar! (es gilt dann jetzt sicherlich)
  - aber wenn die betrachtete Eigenschaft nicht stabil ist, was dann?
  - wer garantiert eigentlich, dass eine Eigenschaft (wirklich) stabil ist?

# Ein Schnappschussalgorithmus



Prozesse, Nachrichten: *schwarz* oder *rot*  
 Schnappschusssaugenblick: *schwarz* --> *rot*  
 (dann: lokalen Zustand dem Initiator melden)  
 Prozess wird *rot*, wenn a) Aufforderung erhalten,  
 b) Erhalt einer roten Nachricht

Beh.: Schnappschuss ist *konsistent*.  
 Bew.: Keine "Nachricht aus der Zukunft"

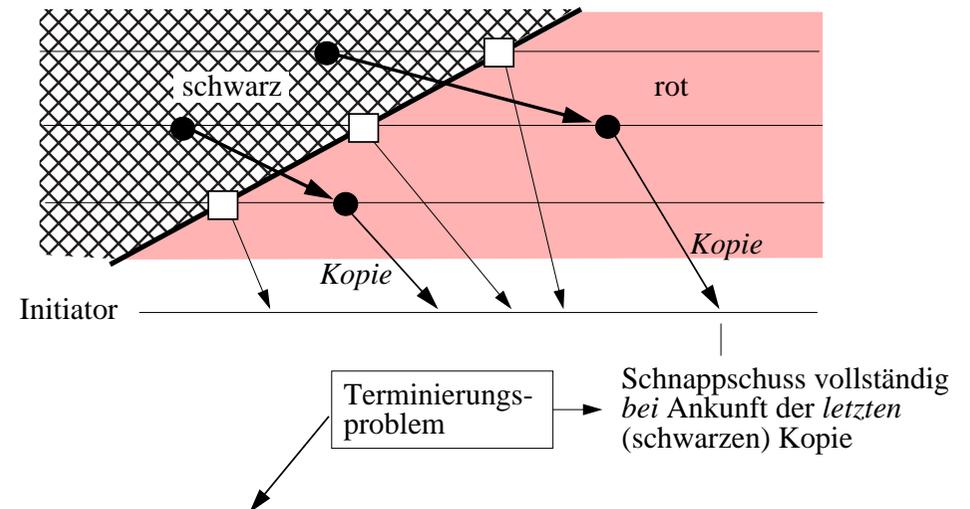
*Nachrichten*, die *unterwegs* sind?

- *Schwarze* Nachrichten, die bei *rot* ankommen
- Sende *Kopie* davon an Initiator
- Problem: Wann *letzte Kopie* dort eingetroffen?

# Schnappschussalgorithmus: Nachrichten

## - In-Transit-Nachrichten?

- schwarze Nachrichten, die von einem roten Prozess empfangen werden
- Sende (bei Empfang) eine Kopie davon an den Initiator
- Problem: Wann hat der Initiator die letzte Kopie erhalten?



## - Z.B. "Defizitzähler" als Teil des konsistenten Zustands

- zähle gesendete und empfangene schwarze Nachrichten
- globale Differenz = Anzahl zu erwartender schwarzer Kopien

## - Wellenalgorithmus als "Basisalgorithmen" --> unterschiedliche Schnappschussalgorithmen

### - FIFO-Kommunikation keine Voraussetzung

### - Letzte In-transit-Nachricht abwarten

- das kann aber lange dauern; geht es nicht auch schneller? (ja! wie?)

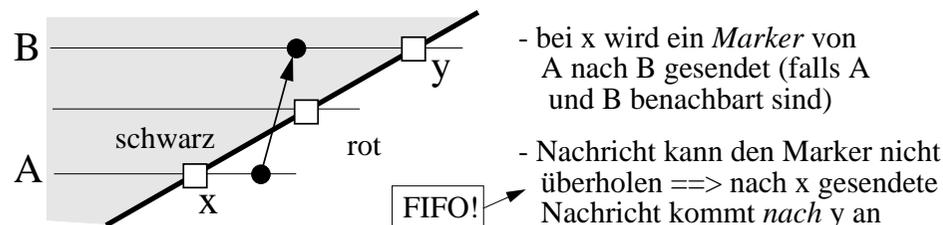
### - "Repeated snapshot": Farben vertauschen

# 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

- $\neg \exists$  Nachricht aus der Zukunft  $\implies$  Schnitt ist *konsistent*  
- vgl. auch frühere Ausführungen zu "virtuell gleichzeitiges Markieren"



# 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*

- *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-Algorithmus statt Flooding)

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

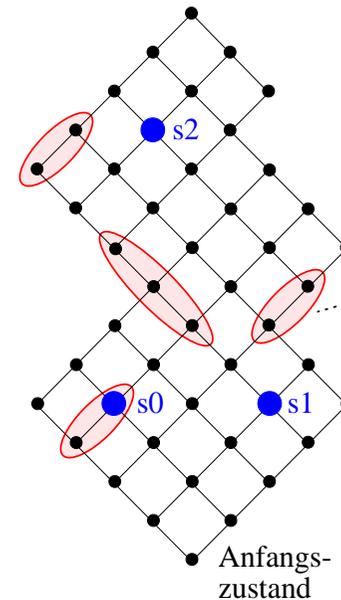
- Wieso ist im ersten Fall der Kanalzustand die leere Folge? Hat jeder so gewonnene globale Zustand einige leere Kanäle?

# 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!



Menge der konsistenten Zustände der Berechnung, geordnet entsprechend der "später-Relation" (entspricht der Präfixbeziehung)

Eigenschaft sei in diesen Bereichen gültig (wird jedoch von s1 oder s2 nicht "entdeckt", allenfalls von s0)

wenn dies in einem Zustand gilt, dann auch in allen späteren

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

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