Various Characterizations of Synchronous Communications

- Question: are they all equivalent?
- Note: some characterizations are informal or less formal than others

---

1) Best possible approximation of *instantaneous* communications

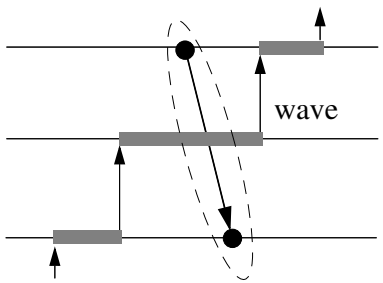
(i.e., without clocks, it is not possible to prove that a message was not transmitted instantaneously)

2) Space-time diagrams can be drawn such that *all message arrows are vertical*

3) *Communication channels* always appear to be *empty*

(i.e., messages are never seen to be in transit)

4) Corresponding *send-receive events* form one *single atomic action*

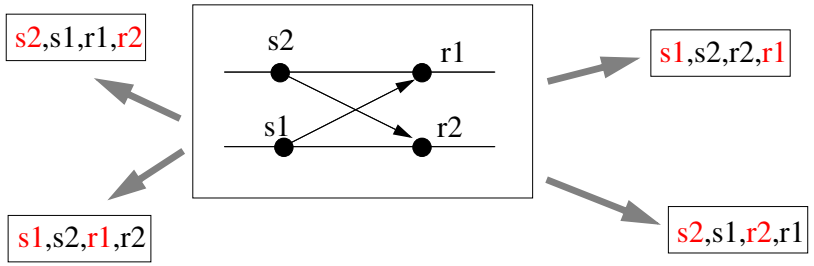


- But what exactly does "atomic" mean?
- Does the combined event happen before or after the wave? Should this be possible with synchronous communication?

set of all events with the causality relation

which belong to the same message

5) ∃ *linear extension* of (E, <) such that ∀ corresponding communic. events s,r: *s is an immedate predecessor of r*



- The example has 4 different linearizations: in all of them a pair of corresponding send-receive events is separated by other events - hence this computation cannot be realized synchronously
- Motivation: corresponding events form a single atomic action

6) Define a (transitive) *scheduling relation* '<' on messages:

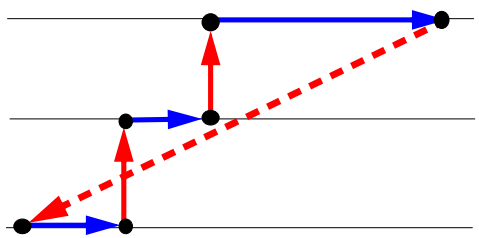
m '<' n iff send(m) < receive(n)

The graph of '<' must be *cycle-free*

this is the classical causality relation on the set of events

- Then whole messages (i.e., corresponding send-receive events s, r) can be scheduled at once (s before r), otherwise this is not possible

7) *No cycle* is possible by moving along *message arrows in either direction*, but always from left to right on process lines



- *Interpretation*: ignoring the direction of message arrows ==>
  - send / receive is "symmetric"
  - "identify" send / receive

- If such a cycle exists ==> no "first" message to schedule
- If no such cycle does exists ==> message schedule exists

8) Synchronous causality relation << is a *partial order*

Definition of << : compare this to the classical causality relation!

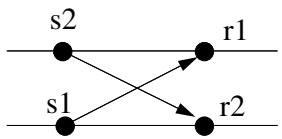
1. If a before b on the same process, then a << b
2. x << s iff x << r (“common past”)
3. s << x iff r << x (“common future”)

   for all corresp. s, r and for all events x
4. Transitive closure

Interpretation: corresponding s, r are not related, but with respect to the synchronous causality relation they are "identified"

they have the same past and future

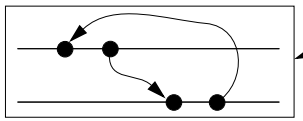
Example:



a) s1 << r2 (1)
b) r1 << r2 (a, 3)
c) s2 << r1 (1)
d) r2 << r1 (b, 3)

cycle, but r1 ≠ r2 !

- Compare this characterization to characterization 6
- Why is the definition of << sensible?

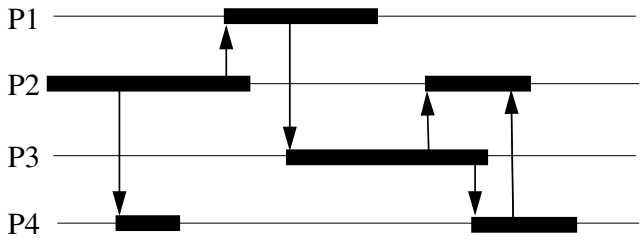


This is something where the classical causality relation < is not a partial order, therefore it is not even realizable as a computation with *asynchronous* messages!

Further reading: http://www.inf.ethz.ch/vs/publ/papers/syn_asy.pdf

Terminierungserkennung bei synchroner Basiskommunikation

- Nachrichtenpfeile senkrecht in Zeitdiagrammen



- Abstrakte Basisberechnung mit zwei Aktionen modellieren:

$state_p$ mit Werten *aktiv* oder *passiv*

X_p: {$state_p$ = *aktiv*}
  $state_q$:= *aktiv* (* "atomares" aktivieren *)

I_p: $state_p$:= *passiv*

Nachricht hier unerheblich...

- Aufgabe: Für dieses abstrakte Modell einer Basisberechnung einen überlagerten Kontrollalgorithmus angeben

- Aktion X_p grenzt an Telepathie...

Verhaltensmodelle vert. Anwendungen

(--> Definitionen Terminierung)

Nachrichtengesteuertes Modell ("Transaktionsmodell"):

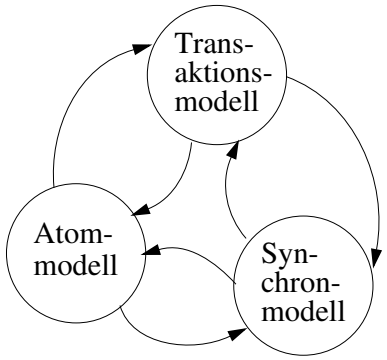
- Alle Prozesse passiv und keine Nachricht unterwegs

Atommodell:
(Prozesse sind immer passiv)

- Keine Nachricht unterwegs

Synchronmodell:
(Nachrichten haben Laufzeit 0)

- Alle Prozesse passiv



- Lassen sich die jeweiligen Lösungsalgorithmen für ein anderes Modell verwenden / transformieren?

  Aufgabe: Man gebe jeweis geeignete Transformationen an!

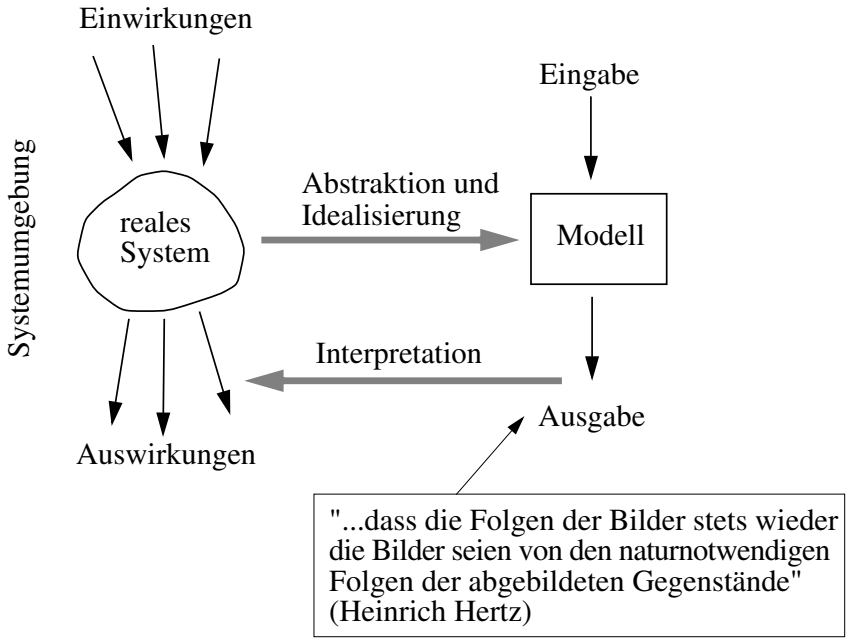
- Was ist das "richtige" Modell?

  - Hängt von der Anwendung ab!
  - Hängt ggf. von Abstraktionsniveau und Sichtweise ab!
  - Ein Modell lässt sich ggf. *realisieren* (Unterschied zu Naturwissenschaften!)
    - z.B. Asynchronität "maskieren" und eine virtuell synchrone Sicht liefern
    - z.B. Aktionen eines Prozesses wechselseitig ausschliessen

Modelle und Modellierung - ein kurzer Exkurs

- Modell = *vereinfachtes* Abbild der komplexen Realität

soll sich bzgl. *relevanter* Aspekte (=?) analog zur Realität darstellen oder verhalten



- Abstraktion --> Reduktion, Vergröberung!

- Modelle dienen u.a. dem Begreifen der Wirklichkeit
  - in diesem Sinne bereits prähistorische Höhlenmalereien
  - Spielen von Kindern
  - Voraussetzung für planvolles Handeln

- Durchspielen am Modell --> *Simulation* ("Modellexperiment")

Heinrich Hertz (1857 - 1894)

> Wir machen uns unsere Scheinbilder oder Symbole der äusseren Gegenstände, und zwar machen wir sie von solcher Art, dass die denknotwendigen Folgen der Bilder stets wieder die Bilder seien von den naturnotwendigen Folgen der abgebildeten Gegenstände... so können wir an ihnen, wie an Modellen, in kurzer Zeit die Folgen entwickeln.
>
> -- Heinrich Hertz: Einleitung zu "Prinzipien der Mechanik", 1894

- Arbeitsgebiete: Funkenentladung, Induktion, Elektrodynamik, Thermodynamik
- geboren in Hamburg 1857
- 1877 immatrikulierte sich Hertz an der Technischen Hochschule München, kümmerte sich aber wenig um den Vorlesungsbetrieb, sondern besuchte mehr die Museen und Theater
- Wechsel nach Berlin, Physikstudium unter Helmholtz und Kirchhoff
- 1879: Goldmedaille der Humboldt Universität für die Lösung einer Preisaufgabe (Nachweis einer trägen Masse von elektrischen Selbstinduktionsströmen)
- Promotion "Über die Induktion in rotierenden Kugeln"
- Assistentenstelle bei Helmholtz
- 1883 Habilitation ("Versuche über die Glimmentladung") an der Uni Kiel
- 1885 Ruf an die TH Karlsruhe
- 1886 Heirat mit Elisabeth Doll, der Tochter eines Kollege; zwei Töchter
- 1889 Professor für Physik in Bonn
- gestorben 1894 an einer Sepsis, noch nicht einmal 37-jährig

Beispiele für Modelle

- Spielzeugeisenbahn ("Modell"eisenbahn)
- Planetarium, Globus, Strassenkarte
- Sandkasten
- Architekturmodell ← bereits in der Antike: Holzmodelle von geplanten Bauwerken
- Flugsimulator
- Training von Astronauten
  - Zentrifuge (Beschleunigung)
  - Parabelflüge (Schwerelosigkeit)
  - isländische Kraterwüste (Mondlandschaft)
- Windkanalmodelle (--> "digitaler Windkanal")
- Differentialgleichungssystem
- Schach und andere Spiele
- Puppe (“Dummy”)
- Ratte ("Tiermodell")
- ...

Korrektes Modellieren ist eine "Kunst":

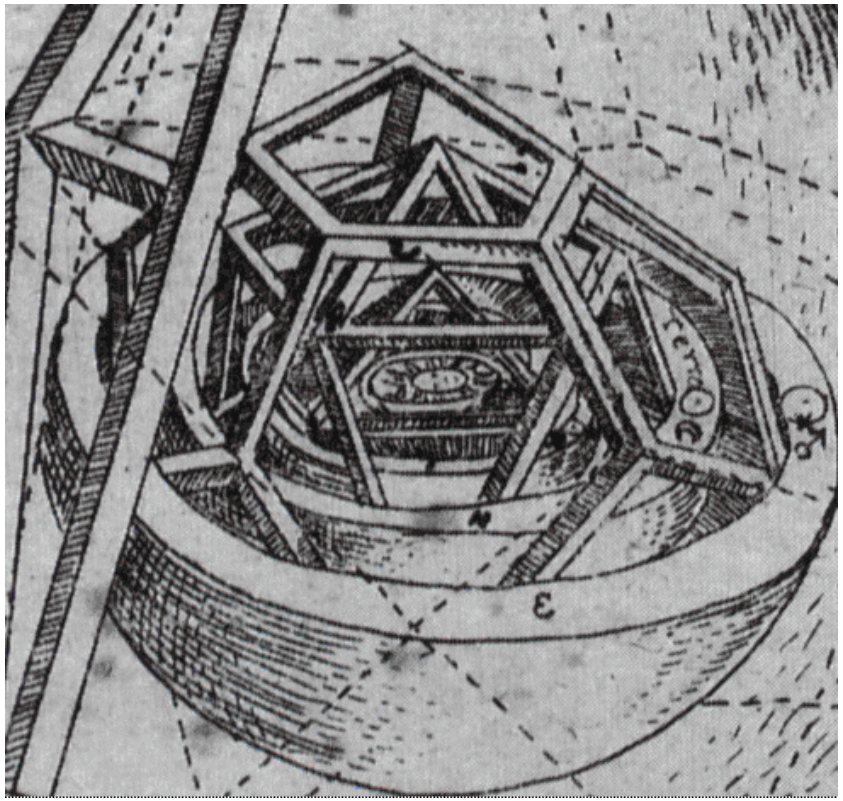
Welche Aspekte werden beibehalten? Wovon wird abstrahiert?

- Konkrete Modelle
  - z.B. massstabsgerechte Verkleinerung
  - z.B Simulation hydraulischer Strömungssysteme durch elektrische Schaltkreise
- Abstrakte Modelle (--> "Computersimulation")

Ein wirklichkeitsfremdes Modell des menschlichen Skeletts (Persien, 14. Jhd.)

Ein Modell der Planetenbahnen aus verschachtelten platonischen Körpern

(J. Kepler, 1597)



Zweck der Modellierung

- Zweck von Modellierung bei der *Simulation der* (ggf. prognostizierten) *Wirklichkeit*

  durch Experimente am Modell

  - Optimierung (z.B. Bestimmung von Engpässen)
  - Entscheidungshilfe (Auswahl von Entwurfsalternativen)
  - Prognose (z.B. Wetter)
  - Validierung (z.B. geplanter Mikroprozessor)
  - Theorienbildung (z.B. kognitive Vorgänge)
  - Animation, Erklärung (pädagog. Hilfsmittel, "Demonstrationsmodell")

- Es geht um das Erkennen und Beherrschen der Realität

- Geschichte der Modellierung und Simulation

  - Höhlenmalereien, Landkarten, Globus
  - Formalisierung der Physik (Differentialgleichungen)
  - Sandkastenspiele der Militärs (Schach)
  - Computer ab ca. 1950 (zunächst militärische und ökonomische Anw.)

The history of mankind is a history of model building
[Rivett: Principles of Model Building, John Wiley, 1972]

The craft of modelling is central to the way in which we understand the world arround us
[Kreutzer: System Simulation, Addison-Wesley]

Implementierung von Modellen

- Im Unterschied zu den *Naturwissenschaften* geht es in der *Informatik* oft nicht um die möglichst detailgetreue Nachbildung der Wirklichkeit, sondern um die *Implementierung* eines Modells als "ausgedachte Wirklichkeit"

- Weil das ausgedachte Modell "schön" ist, z.B.:
  - als hätte man einen sehr grossen Hauptspeicher (--> "virtueller Speicher")
  - als ob ich einen Rechner exklusiv für mich alleine hätte (--> "virt. Maschine")
  - als würden nie Bits auf einer Leitung verloren gehen
  - als ob Nachrichten keine Zeit bräuchten
  - als ob ich eine tolle Figur wäre (IRC, MUD, Avatar...)
  - als ob...

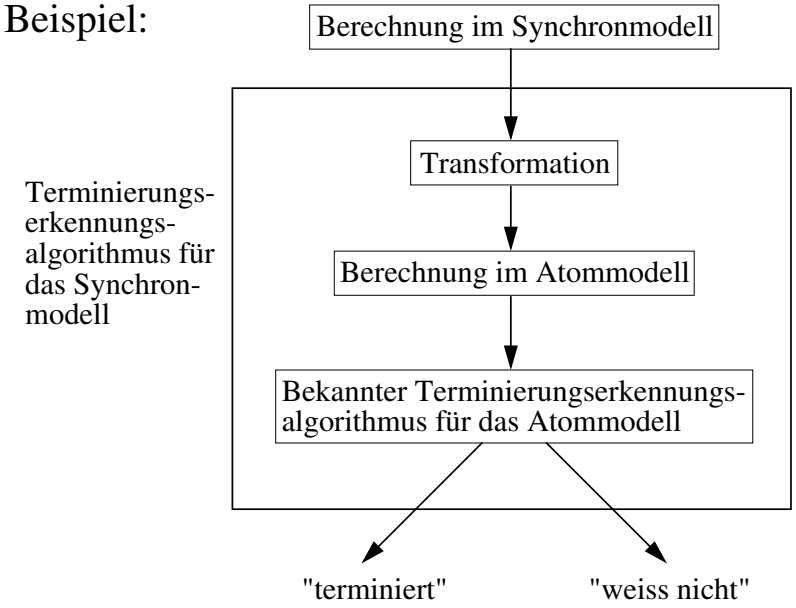
- Das heisst: Man erschafft sich eine "Wirklichkeit"
  - künstliche Wirklichkeit = virtuelle Realität

Modelltransformation

- Terminierung lässt sich definieren / lösen im Transaktionsmodell, Atommodell und im Synchronmodell
  - Problem: Wie wird ein Lösungsalgorithmus (z.B. zur Erkennung der verteilten Terminierung) für ein anderes Modell adaptiert?

- Idee: Transformation des *Modells*, nicht der Algorithmus

- Beispiel:



- Transformation muss natürlich "korrekt" sein: Wenn die transformierte Berechnung beendet ist, dann ist auch die Originalberechnung beendet (--> safety); wenn die Originalberechnung beendet ist, dann ist schliesslich auch die transformierte Berechnung beendet (--> liveness)

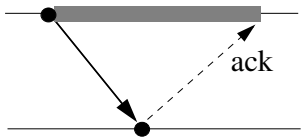
Ein Beispiel für eine Modelltransformation

Gegeben:

- Berechnung im Atommodell
- Terminierungsalgorithmus für Synchronmodell

statt Atommodell

Was tun? --> Einführung von "passiv" und "aktiv"
(Modelltransformation)



- Sende ack zurück bei Empfang einer Nachricht
- Sender wird "aktiv" beim Senden, "passiv", wenn alle Nachrichten quittiert

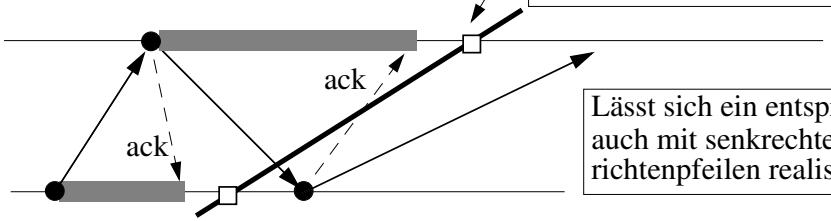
Dann gilt: Alle "passiv" ==> keine Nachricht unterwegs

stellt Terminierungsalgo. für Synchronmodell fest

ist Terminierungsdefinition für das Atommodell

Aber was ist damit?:

Zu jeder (bekannten) Nachricht wurde ein ack empfangen und alle sind passiv. Terminiert?

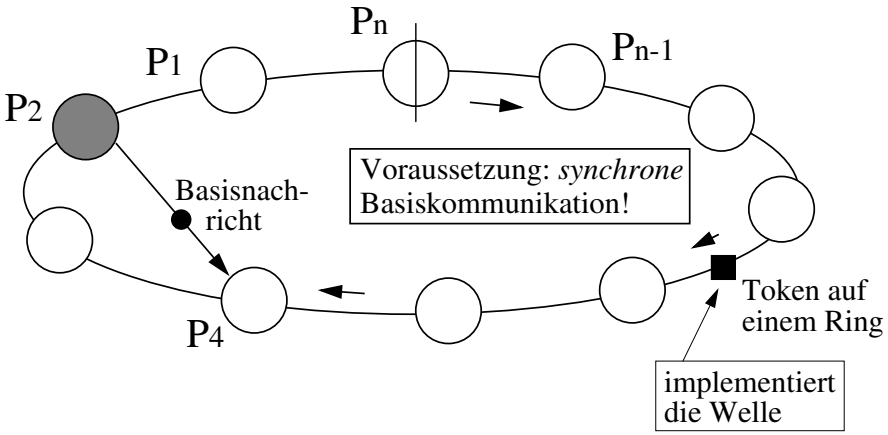


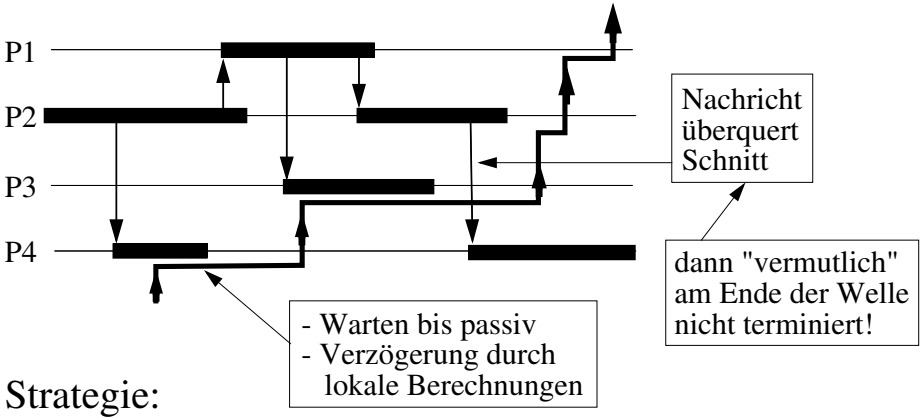
Lässt sich ein entspr. Szenario auch mit senkrechten Nachrichtenpfeilen realisieren?

- "Alle sind passiv" (entlang einer schiefen Schnittlinie!) ist kein korrektes Terminierungskriterium im synchronen Fall!
- Genausowenig wie "alle passiv und alle acks angekommen" im Transaktionsmodell!

Der DFG-Algorithmus

- Dijkstra, Feijen, Van Gasteren, 1983, Inf. Proc. Lett.
- Zur Terminierungserkennung bei syn. Basiskommunikation





Strategie:

- Prozesse und Token können schwarz oder weiss sein
- Prozess wird schwarz, wenn er an einen Prozess mit einer höheren Nummer etwas sendet
- Welle testet, ob ein Prozess schwarz ist und färbt Prozess ("auf der Rückflanke") weiss
- Terminiert, wenn alle weiss

DFG-Algorithmus: Verhaltensregeln

Regel 1: Ein Prozess, der eine Basisnachricht [an einen Prozess mit einem höheren Index] sendet, wird schwarz.

Regel 2: Wenn Prozess P_n passiv ist, kann er den Terminierungstest initiieren, indem er ein weisses Token an P_{n-1} sendet.

Regel 3: Ein aktiver Prozess behält das Token, bis er passiv wird.

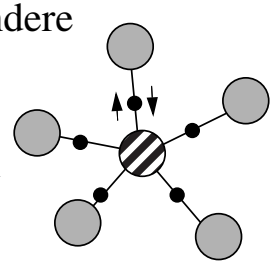
Regel 4: Ein passiver Prozess P_i ($i \neq n$), der das Token hat, reicht ein schwarzes Token weiter an P_{i-1}, wenn er oder das Token schwarz ist, ansonsten reicht er ein weisses Token weiter.

Regel 5: Ein Prozess, der das Token weiterreicht, wird weiss.

Regel 6: Wenn Prozess P_n ein weisses Token erhält, meldet er *Terminierung*.

Regel 7: Wenn Prozess P_n ein schwarzes Token erhält, startet er eine neue Runde.

- Korrektheit ? (Safety; Liveness)
- Muss das Token mittels synchroner Kommunikation propagiert werden?
- Wieso klappt der Algorithmus nicht für asynchrone Basiskommunikation?
- Worst-case Nachrichtenkomplexität? "Detection delay"?
- Muss der Initiator eindeutig sein? Mehrere Initiatoren?
- Statt Kontrollring andere Realisierung der Kontrollwelle?
- Welche Farbe haben die Prozesse initial?



z.B. Stern mit einem zentralen Initiator, der *parallel* Token zu den Prozessen schickt, um deren Farbe zu ermitteln