TECHNISCHE UNIVERSITÄT MÜNCHEN

Kompositionalität

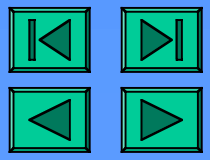
Perlen der Weisheit

Bernhard Schätz

Modularität

Jahr	Sprache	Eigenschaften
1957	FORTRAN	if, loop, integer, real, ...
1960	COBOL	arrays, records, strings, ...
1965	Simula	objects, pointers, coroutines,...
1971	Pascal	case,safe typing,...
1979	ADA	package, exception, task,...
1995	Java	virtual machine,...

- **Strukturierte Programmierung**
  - **Kontrollfluss: Sequenz, Auswahl, Wiederholung**
  - **Abstraktion: Abstrakte Typen, Prozeduren**
- **Modularität:**
  - **Kapselung, Informationsverbergen**
  - **Abhängig vom Schnittstellenbegriff (Datum/Aktion)**

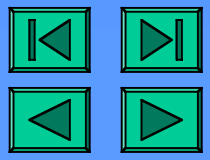


Co-Routinen

```
Cobegin
{ True }
x = Ø
 { x < 5 }
 while x < 5 do
 { x < 5 }{ x = k -1 }
 x = x + 1
 { x <= 5 }{ x = k }
 od
{ x = 5 }
coend
```

```
Cobegin
 while x > Ø do
  x = x - 1
 od
coend
```

- **Ausgangssituation:**
  - **Modulares Programmieren**
  - **Axiomatische Semantik für proz. Sprachen**
  - **Entwicklung paralleler Programme**
- **Problem: Verlust der Modularität**

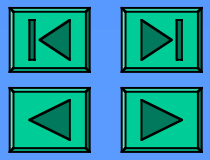


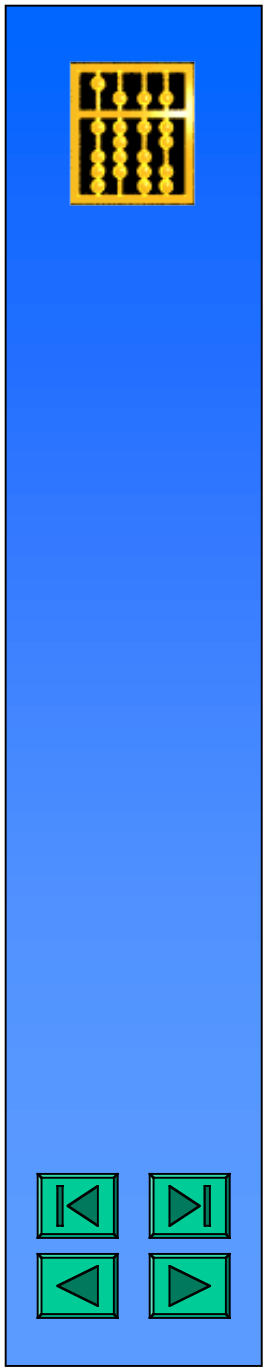
„Freedom of Interference“

```
Cobegin
{  True }
x  =  Ø
  { x  <  5  }
  while x  <  5  do
  { x  <  5  }{  x  =  k  -1  }
  x  =  x  +  1
  { x  <=  5  }{  x  =  k  }
  od
{ x  =  5  }
coend
```

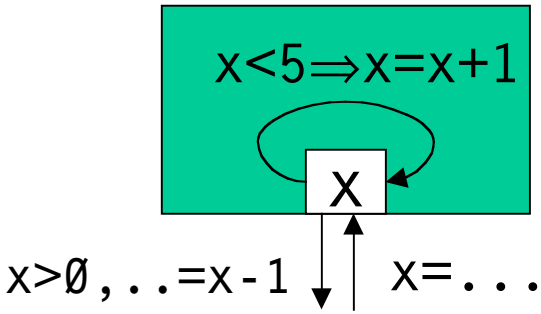

```
Cobegin
  while x  >  Ø  do
   { x  >=  Ø  &  x  =  k  +  1  }
   x  =  x  -  1
   { x  >  Ø  &  x  =  k  }
  od
coend
```

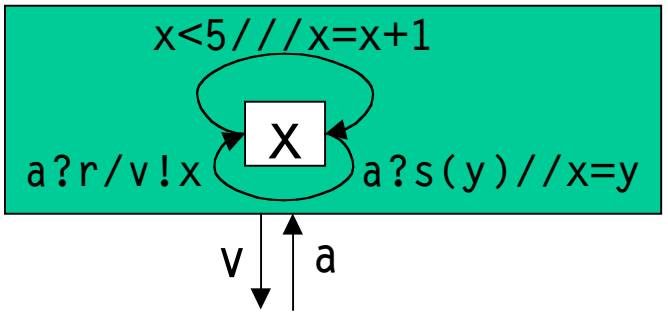
- **Idee: Zerlegen der Beweisverpflichtung**
  - **Verfahren: Hoare/Owicki-Gries**
    - **Führe Nachweis für Komponente**
    - **Zeige: Umgebungsschritte beeinflussen Komp. nicht**
  - **Allgemein:**
    - **Umgebungsunabhängige Spezifikation**
    - **Eingeschränkt auf Umgebung**





Reaktivität vs. Parallelität



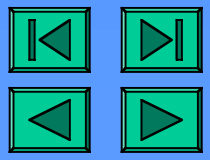


- **Parallelität:**
  - **Module mit unabhängiger Abarbeitung**
- **Reaktivität**
  - **Komponenten mit ansteuerbaren Aktionen**
- **Paradigma:**
  - **Übertragung des Modulprinzips**
  - **Schnittstelle: von Daten zu Aktionen**

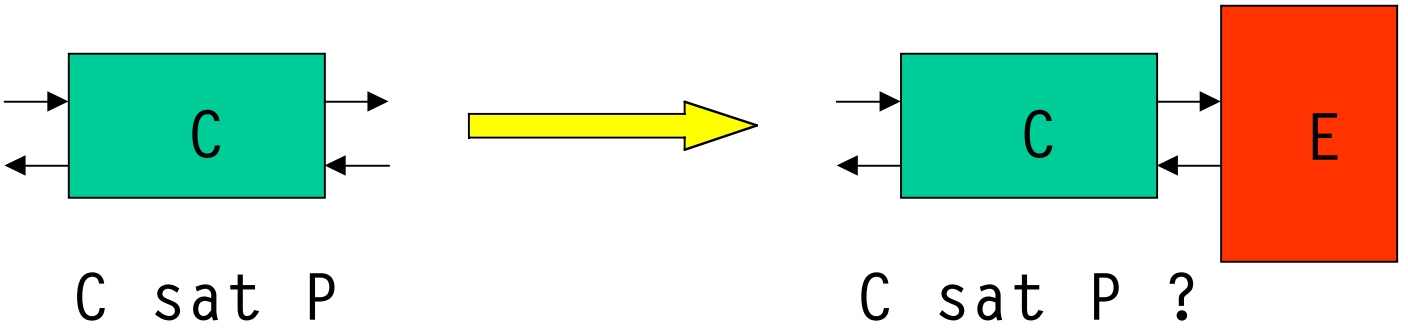
Schwerpunkte

	Sequentiell	Reaktiv
Verhalten	Ein/Ausgabe	Abläufe
Partiell korrekt	Ausgabe korrekt	Sicherer Ablauf
Problemsystem	Nichttermination	Verklemmung
Vollständ. korr.	Termination	Lebendigkeit

- **Parallele Systeme**
  - **Zwischen sequentiell und reaktiv**
  - **Parallelisierung sequentieller Berechnungen**
  - **Vermeidung von Kommunikationsaufwand**
  - **Näher am sequentiellen Modell**
  - **Keine Kapselung des Verhaltens**
  - **„Thread“ vs. Prozess/Task**



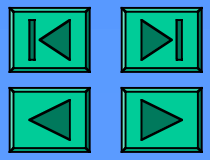
Kompositionalität 1



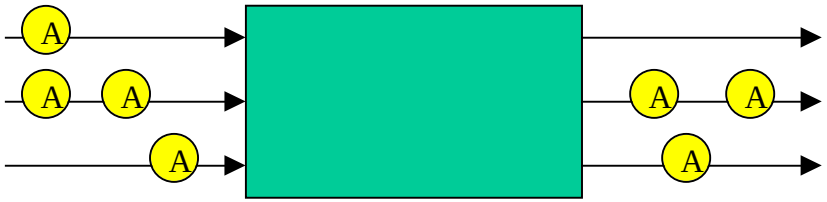
- **Speziell:**
  - **Modularisierung des Entwurfs:**
    - **Realisierung einer Komponente C mit Eigenschaft P**
    - **Einbettung in beliebige Umgebung E**
    - **Einbettung erhält Eigenschaft P**
  - **Freedom of Interference:**
    - **„Eingebaut“ in Modell**

Denotationelle Semantiken

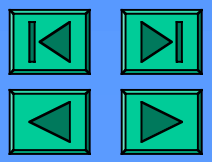
- Grundlagen:
  - Scott‘sche Bereichstheorie zur LCF
  - Ansatz:
    - System als mathematisches Objekt (Ein/Ausgabe)
    - Definierte Schnittstelle (Funktion) : Werte
    - Ohne Umgebungseinschränkung
    - Komposition über Schnittstellen: Applikation
  - Reaktive Systeme
    - Mathematisches Objekt (Historien/Interaktionen)
    - Definierte Schnittstelle: Interaktionen
    - Ohne Umgebungseinschränkung (Enabled, Blocking)
    - Komposition über Schnittstelle: Projektion

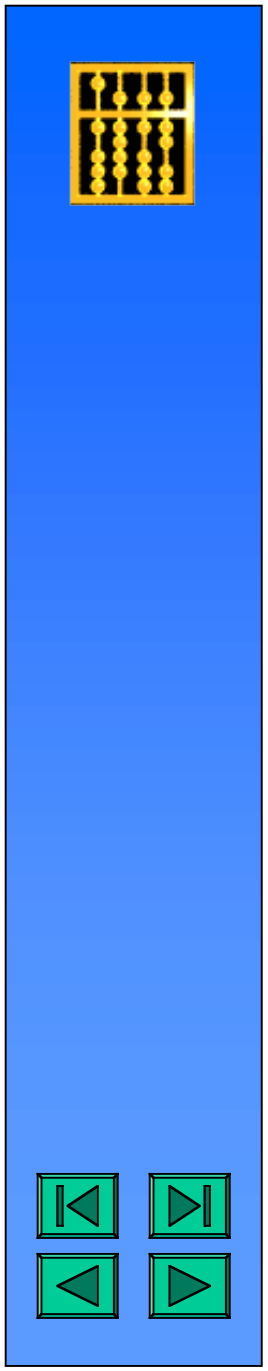


Asynchrone Systeme

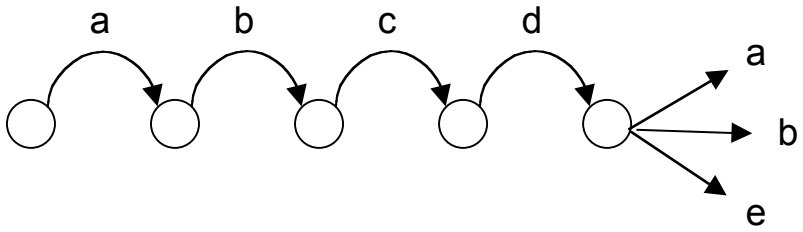


- **Semantisches Paradigma:**
  - **Kommunikation:**
    - **Gerichtete Kanäle (Ein/Ausgabe unterscheidbar)**
    - **Eingabevollständig (Eingabe nicht blockierbar)**
  - **Modellierung:**
    - **Atomare Interaktionen (Alphabet)**
    - **Verhalten = Sequenzen von Interaktionen**
  - **Klassische Modelle**
    - **Spurmodelle (z.B. Dill)**
    - **Funktionsmodelle (z.B. Kahn)**
    - **Automatenmodelle (z.B. Lynch/Tuttle)**



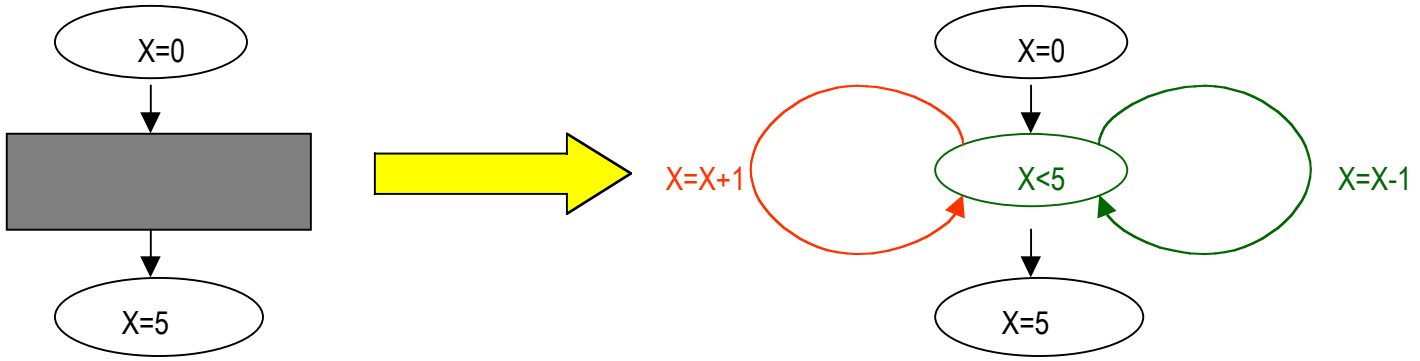


Synchrone Systeme

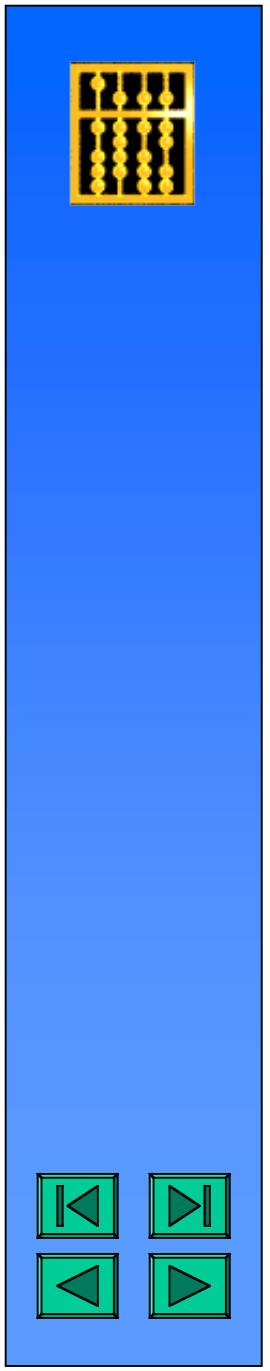


- **Semantisches Paradigma:**
  - **Kommunikation:**
    - **Über gemeinsame Ereignisse**
    - **Zusätzlich: Blockierende Kommunikation**
  - **Modellierung:**
    - **Atomare Interaktionen (Alphabet)**
    - **Verhalten = Sequenzen von Interaktionen**
  - **Klassische Modelle**
    - **Failuresets (z.B. Hoare)**
    - **Readinesssets (z.B. Olderog)**

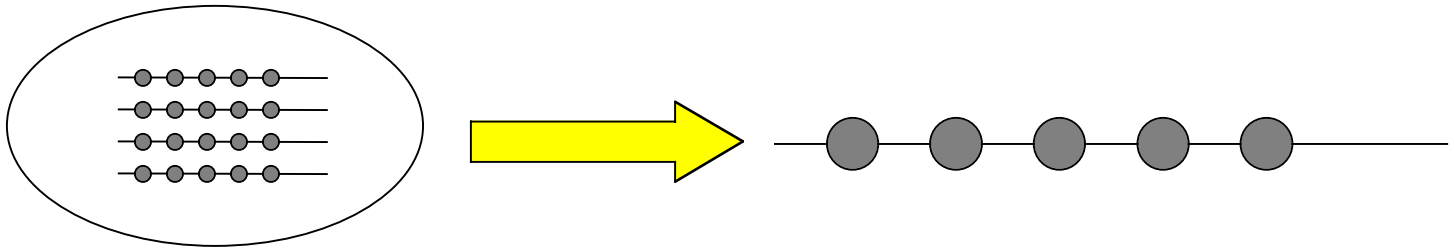
„Freedom of Interference“



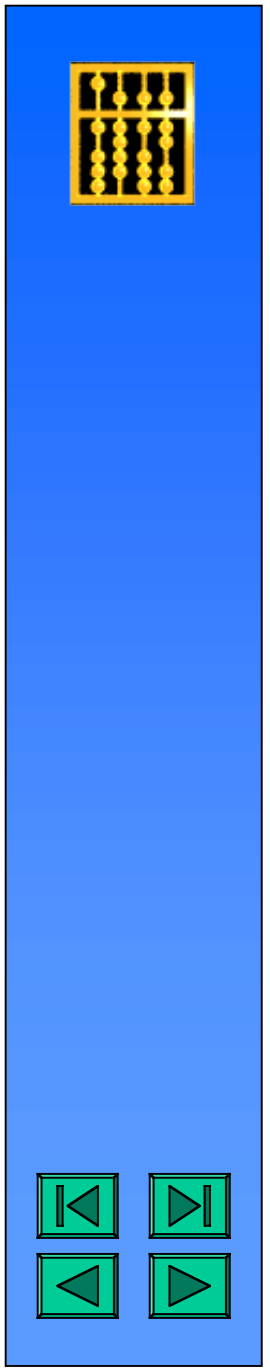
- Owicki/Gries:
  - Nachweisung der Einmischungsfreiheit
- Semantiken reaktiver Systeme
  - „Explizitmachen“ der Umgebungseinmischung
    - Explizite Darstellung der Systementwicklung (Traces)
    - Explizite Darstellung der Interaktion (Tracelemente)
  - Resultat:
    - Mögliche Umgebungsaktionen a priori erfasst (total)
    - Kombination mit beliebiger Umgebung möglich



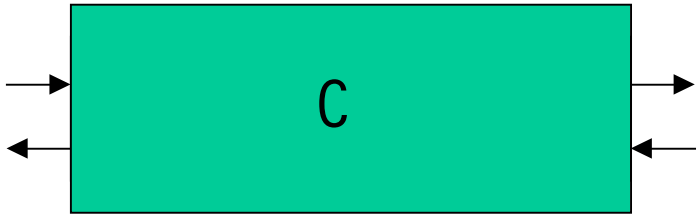
Beobachtungsbegriff



- **Spezifikationen:**
  - **Beschreibung von Eigenschaften:**
    - **Zusammenhängende Beobachtung**
    - **Historie von Interaktionen**
    - **Nur Aussagen über eine gesamte Beobachtung**
  - **Eigenschaften und Spezifikation**
    - **Spezifikation = Aussage über alle Beobachtungen**
    - **Formal: frei/universell quantifiziert über Beobachtung**
    - **Andere Form:**
      - **Existentielle Aussagen: Immer möglicher Ablauf**
      - **Nicht kompositional**

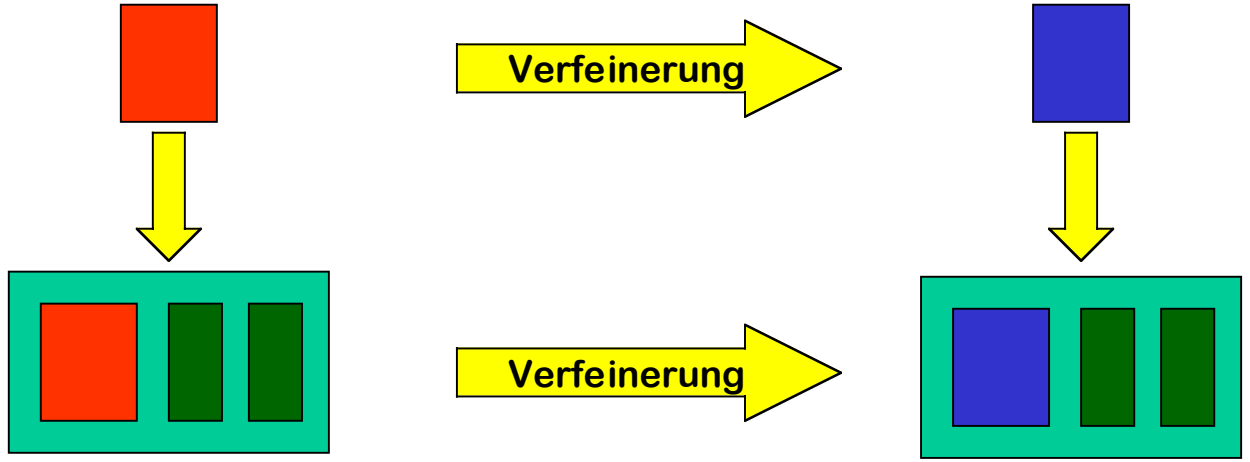


Basisoperationen

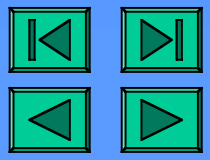


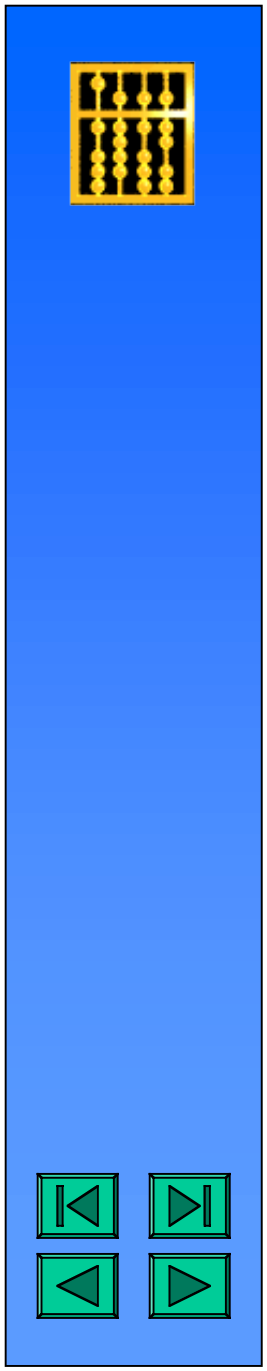
- Operationen:
  - Verbergen
    - Strukturelles Verbergen (Blackbox)
    - Denotationelle Semantiken: Ex.Quantifikation
  - Komponieren
    - Strukturelle Komposition mit Interaktion
    - Denotationelle Semantiken: Konjunktion
    - Eigenschaftserhaltend (Universale)

Kompositionalität 2

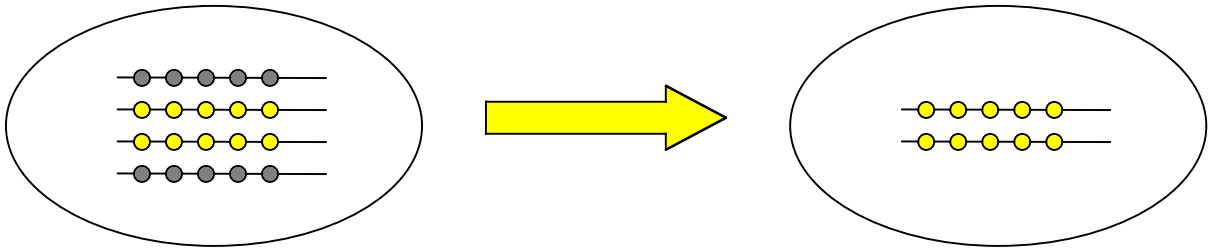


- **Allgemein:**
  - **Verträglichkeit von Verfeinerung (Kongruenz)**
    - **Strukturell (Komponente → Subkomponente)**
    - **Verhalten (Unterspez. → Determinisierung)**
  - **Pragmatisch:**
    - **Komponenten konsistent ersetzen erlaubt!**
    - **Komponente verbessert = System verbessert**

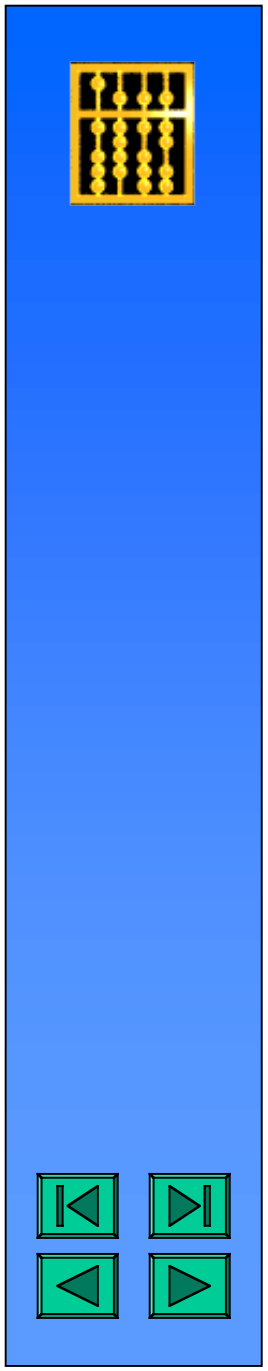




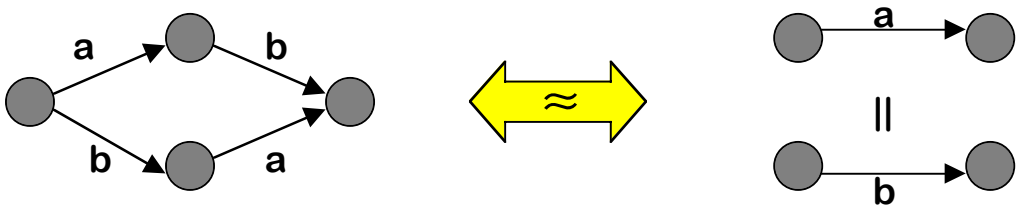
Denotationelle Semantiken



- **Verfeinerungsbegriff:**
  - **Allgemein: Teilmengen/Implikation**
- **Komposition:**
  - **Allgemein: Parallelkomposition/Konjunktion**
- **Kompositionalität:**
  - **Allgemein: Folgt aus Kongruenz der Implikation**



Operationelle Semantiken



- **Operationelle Semantiken**
  - **Beobachtungsbegriff:**
    - **Basierend auf Zuständen und Auswahlmöglichkeiten**
    - **Bisimulation: Vergleich zweier Prozesse**
  - **Verfeinerungsbegriff:**
    - **(Bi)simulation**
  - **Strukturverträglichkeit:**
    - **(Bi)simulationsbegriff ist eine Kongruenzrelation**

Zusammenfassung

- **Modularität**
  - Zerlegung in abgeschlossene Module
    - Schnittstelle
    - Unzugängliche innere Struktur
- **Kompositionalität:**
  - Systemeigenschaften aus Komp.-Eigenschaften
    - Einfach (Konjunktion)
    - Verfeinerungsverträglich (Implikation)
- **Zusammenhang:**
  - Interaktionshistorie als Schnittstellenbegriff
  - Assumption/Commitment-Spezifikationen
  - Einfluss auf Programmiersprachen: Ada, Modula