Modellierung verteilter Systeme

Grundlagen der Programm und Systementwicklung

Sommersemester 2012

Prof. Dr. Dr. h.c. Manfred Broy

Unter Mitarbeit von Dr. M. Spichkova, J. Mund, P. Neubeck

Lehrstuhl Software & Systems Engineering

Teil 4: Struktur- und Verteilungssicht

Systemarchitektur

Komposition, Struktur- und Verteilungssicht: Architektur

- Hierarchische Zerlegung eines Systems in Teilsysteme (*Komponenten*)
  - Top-down
  - Bottom-up
- Komponenten kooperieren (z.B. durch Nachrichtenaustausch)
  - Untereinander
  - Mit der Systemumgebung
- Darstellung des Datenflusses mit *Datenflussmodellen*
  - Graphisch: Datenflussnetz
  - Interaktionsorientiert: Stromverarbeitende Funktionen

Architekturen und verteilte Systeme

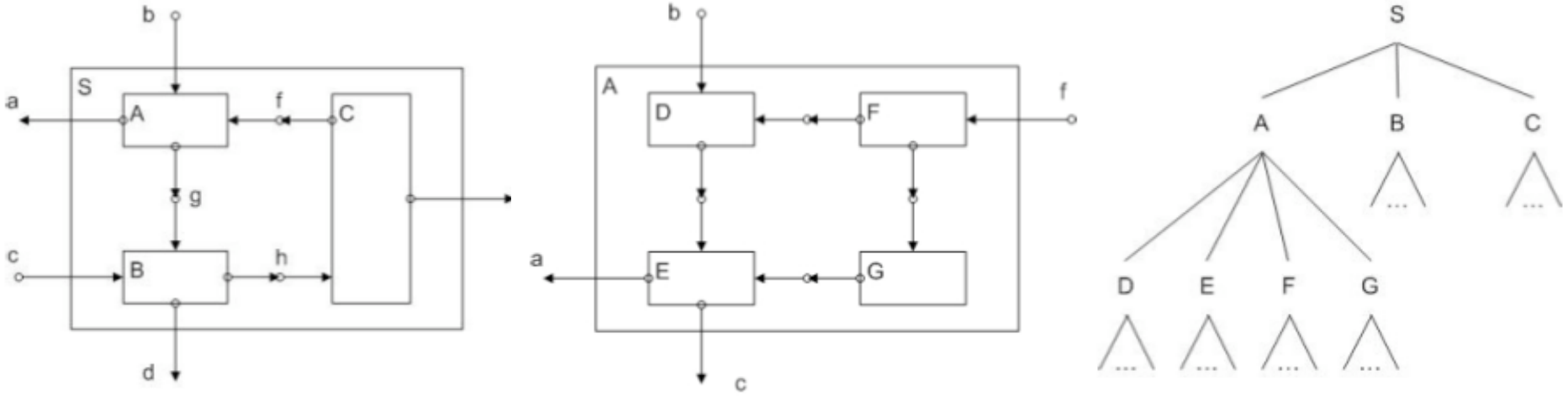
- Verhalten eines verteilten Systems auf Architekturebene
  - Zerlegung in Komponenten
  - Schnittstellenspezifikation der Komponenten
  - Präzise Festlegung der Verbindungen im Sinne des Informationsaustauschs
- Varianten der Modellierung und Beschreibung verteilter Systeme
  - Charakterisierung der Komponenten und deren Schnittstelle
  - Zusammenwirken der Komponenten

Rolle der Architektur im System-Lebens-Zyklus

- Modulares Aufspalten eines Systems in Teilsysteme (Divide and Conquer)
  - Reduzierung der Komplexität
  - Planpause für Entwicklung
  - Übersicht über Lösungsansatz
  - Information Hiding (Verbergen von Implementierungsdetails)
  - Definition von Teilentwicklungsaufgaben - Arbeitsteilige Entwicklung
  - Verteilung auf die Hardware
  - Wiederverwendung von eigenständigen Systemteilen

Architekturen und zusammengesetzte Systeme

- Architektur eines zusammengesetzten Systems beschreibt die Struktur des Systems, gegeben durch
  - Endliche Menge *K* der Komponenten
  - Angabe, welche der Komponenten wie untereinander oder mit der Systemumgebung verbunden sind



Parallele Komposition

- Verteilte Systeme entstehen durch (parallele) Komposition ihrer Teilsysteme
  - Zeitliches nebeneinander - gleichzeitiges Ausführen
  - Verschränktes Ausführen (Interleaving)
- Definition der parallelen Komposition von
  - Zustandsmaschinen mit Ein-/Ausgabe
  - Zustandsmaschinen mit gemeinsamen Speichern
  - ...viele weitere Konzepte
- Zur Komposition müssen
  - syntaktische Schnittstellen der Komponenten zusammenpassen
  - aus dem Verhalten der Teilsysteme das Verhalten des Gesamtsystems ermittelt werden

Formen der Komposition

- Die Komposition von Systemen (genannt Komponenten) zu zusammengesetzten Systemen erfolgt
  - Zusammenfügen:
    Durch die Verbindung der Komponenten und die Festlegung, mit welchen Mitteln die Komponenten zusammenwirken
  - Verhalten:
    Durch die Berechnung der Verhaltensbeschreibung des zusammengesetzten Systems aus den Verhaltensbeschreibungen der Komponenten
  - Kapselung:
    Durch die Festlegung des nach Außen sichtbaren Verhaltens (Schnittstellenbeschreibung des zusammengesetzten Systems)

- Im Vordergrund:
  - Wie die Komponenten beschrieben sind?
  - Wie sie zusammengefügt werden können?
  - Wie das Verhalten des zusammengesetzten Systems sich aus dem Verhalten der Komponenten ergibt?

Beispiele für Systemkomposition

- Zustandsmaschinen
  - Bei Aktionen mit unmarkierten Zustandsübergängen:
    Interleaving mit gemeinsamen Variablen
  - Bei Aktionen mit markierten Zustandsübergängen:
    Komposition über gemeinsamen Aktionen (siehe später)
  - Bei Maschinen mit Ein/Ausgabe:
    Wechselseitiger Austausch der Ein/Ausgaben in „Rückopplungsverbindungen“ (engl. *Feedback Loops*)

- Bei Schnittstellenmodellen für Systeme mit Ein/Ausgabekanälen
  - Verbindung über Kanäle

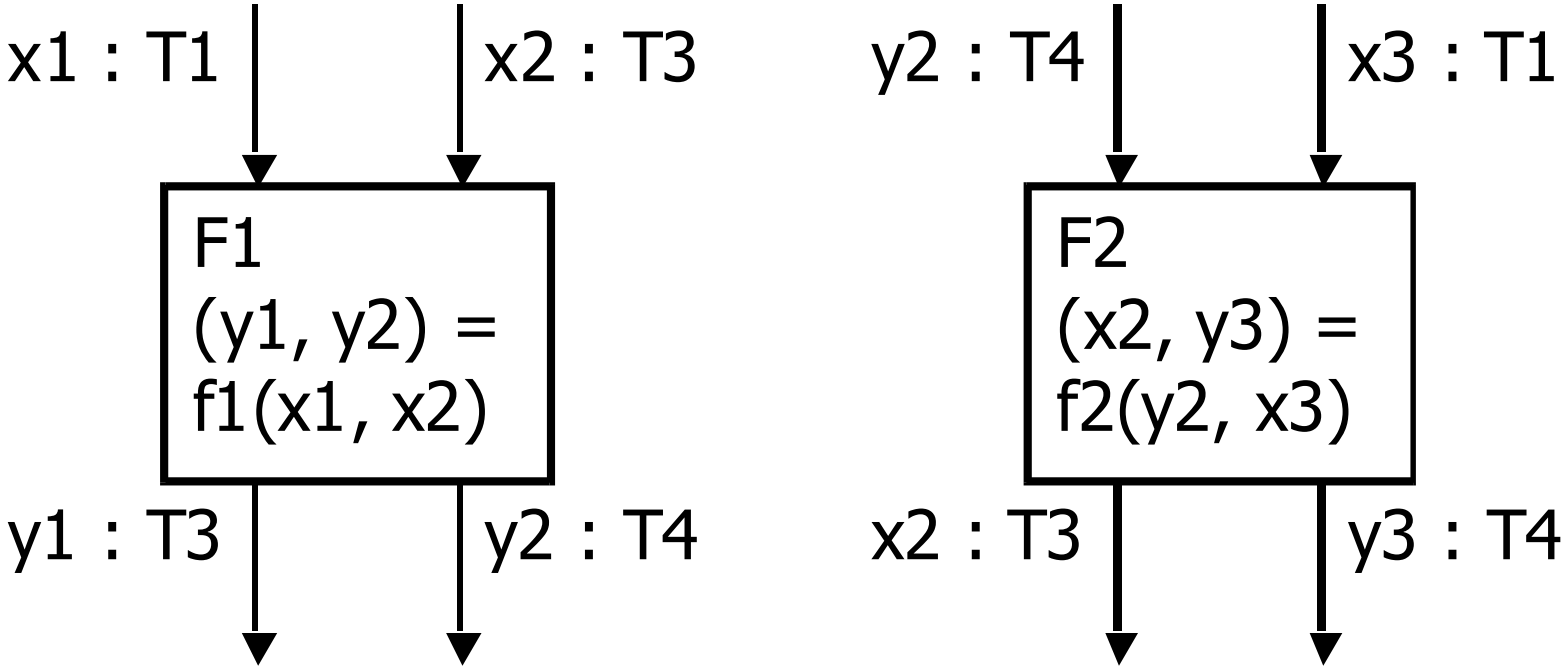
Modularität

- Verhalten eines Systems ergibt sich aus
  - Verhalten der Teilsysteme
  - Zusammensetzung aus den Teilsystemen (Struktur)
- Modellierungsansatz ist **modular**, wenn
  - Verhalten des Systems aus der Komposition des Verhaltens der Teilsysteme ermittelbar ist
- Modularität ist nicht selbstverständlich

Komposition von Systemen mit Ein/Ausgabekanälen

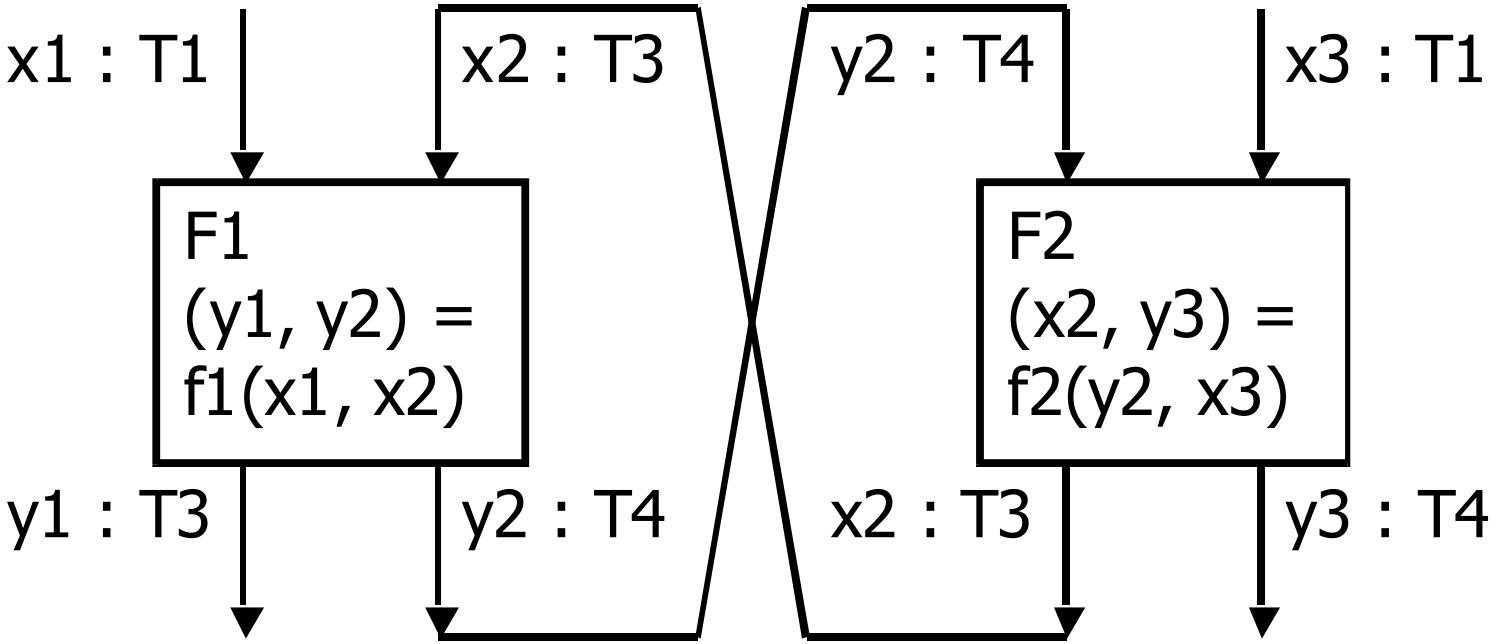
- Verbindung entsprechender Kanäle
  (mit gleichen Identifikatoren und gleichen Datentypen)

- Es entsteht ein Datenflussnetz



Komposition von Systemen mit Ein/ausgabekanälen

- Um Systeme mit Ein/Ausgabekanälen zu komponieren, verbinden wir die entsprechenden Kanäle (mit gleichen Identifikatoren und gleichen Datentypen)
- Es entsteht ein Datenflussnetz

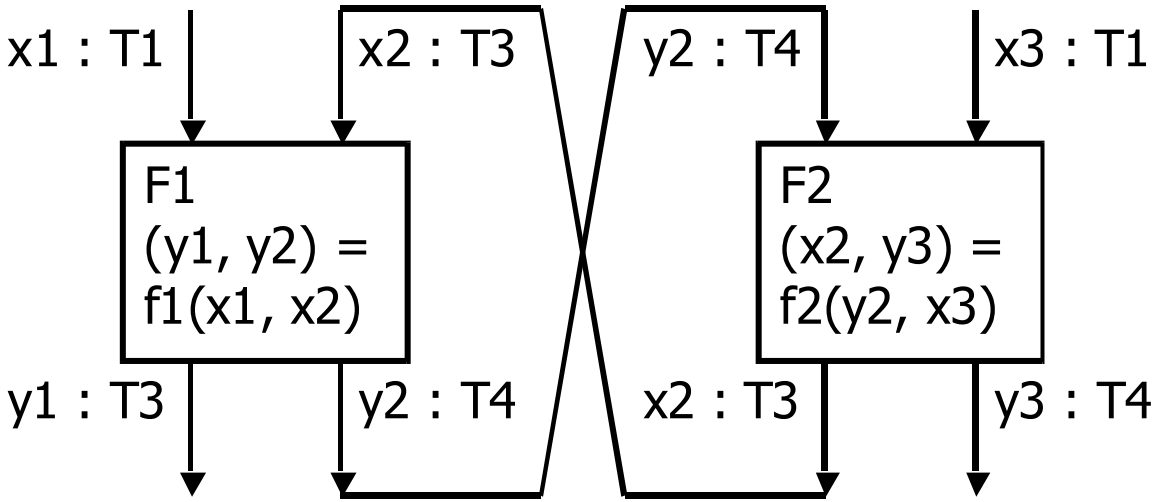


Gleichungssystem

Für gegebene Eingabeströme s1, s3 erhalten wir die Gleichungen

(y1, y2) = f1(x1, x2)

(x2, y3) = f2(y2, x3)



Kritische Fragen:

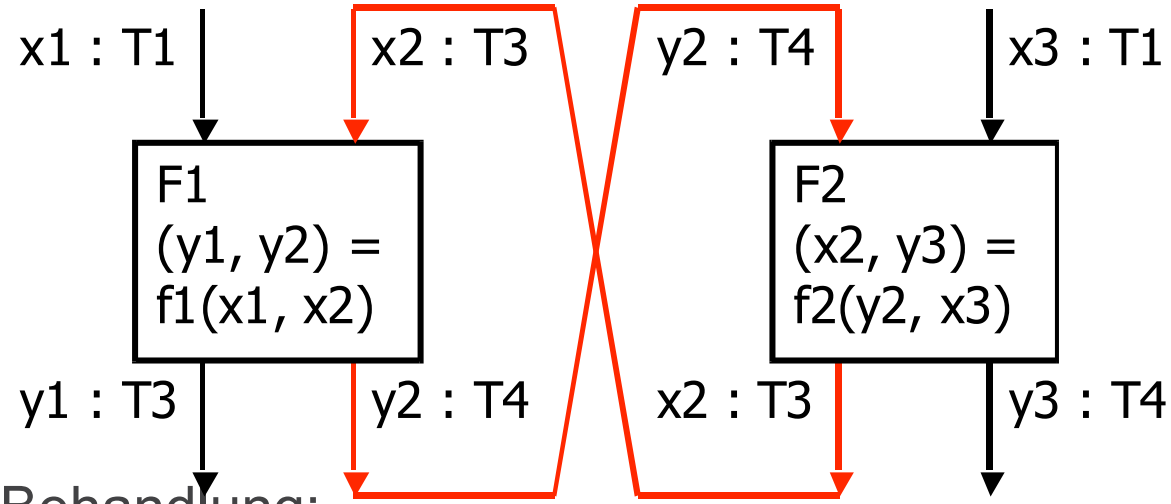
- Gibt es immer eine Lösung für die Gleichungen?
- Ist die Lösung eindeutig?

Rückgekoppelte Kanäle

Rückgekoppelte Kanäle x2, y2

(y1, y2) = f1(x1, x2)

(x2, y3) = f2(y2, x3)



erfordern eine besondere Behandlung:

Bestimmte Eigenschaften der Funktionen garantieren

Existenz von Lösungen (Fixpunkte)!

Wie können wir sicherstellen, dass
die Lösungen das Systemverhalten wiedergeben?

Beispiel: Schwache Kausalität: Unverzögerte Rückkopplung

Seien x1, x2, x3, y1, y2, y3 : $IN_+ \rightarrow IN$ Ströme von Zahlen

$(y1, y2) = f1(x1, x2) \Rightarrow \forall\, t \in IN_+ :\ y1(t) = y2(t) = x1(t)+x2(t)+1$

$(x2, y3) = f2(y2, x3) \Rightarrow x2 = y2 \wedge y3 = x3$ *Identität*

Durch die Komposition mit Rückkopplung gelten beide Gleichungen und damit

$$(y1, y2) = f1(x1, x2) \wedge (x2, y3) = f2(y2, x3)$$
$$\Rightarrow$$
$$\forall\, t \in IN_+ :\ y1(t) = x2(t) = x1(t)+x2(t)+1$$

Damit ergibt die Komposition

$$(y1, y2) = f1(x1, x2) \wedge (x2, y3) = f2(y2, x3)$$

eine **inkonsistente Spezifikation**.

Beispiel: Starke Kausalität: Verzögerte Rückkopplung

Seien x1, x2, x3, y1, y2, y3 : $IN_+ \rightarrow IN$ Ströme von Zahlen

$(y1, y2) = f1(x1, x2) \Rightarrow y1(1) = y2(1) = 0 \wedge \forall\, t \in IN_+: \; y1(t+1) = y2(t+1) = x1(t)+x2(t)+1$

$(x2, y3) = f2(y2, x3) \Rightarrow x2 = y2 \wedge y3 = x3$ *Identität*

Durch die Rückkopplung gelten beide Gleichungen und damit

$$(y1, y2) = f1(x1, x2) \wedge (x2, y3) = f2(y2, x3)$$
$$\Rightarrow$$
$$\forall\, t \in IN_+ : \; y1(t+1) = x2(t+1) = x1(t)+x2(t)+1$$

Damit ergibt die Komposition für gegebene Ströme auf den Kanälen x1 und x3 eine induktive Definition für die Ströme auf den Kanälen x2, y1, y2, y3.

Beispiel: Starke Kausalität: Verzögerte Rückkopplung

Aus der Aussage

$$\forall\, t \in \mathbb{N}_+ :\ y1(t+1) = x2(t+1) = x1(t)+x2(t)+1$$

folgt:

$$y1(t+1) = t+\sum_{1 \le i \le t} x1(i)$$

Beweis: Induktion über t:

t = 0: Trivialerweise gilt y(1) = 0

Gelte die Aussage für t‘ < t+1:

$$y1(t+2) =$$

$$x1(t+1)+x2(t+1)+1 =$$

$$x1(t+1)+y1(t+1)+1 =$$

$$x1(t+1)+\ t+\sum_{1 \le i \le t} x1(i)+1 =$$

$$x1(t+1)+\sum_{1 \le i \le t+1} x1(i)$$

Ansätze für rückgekoppelte Kanäle

- Monotone Funktionen
  - Ströme mit Präfixordnung bilden vollständige partielle Ordnung
  - Unendliche Ströme als Grenzwerte (kleinste obere Schranken) von Ketten endlicher Ströme
  - f1 und f2 präfixmonotone Funktion auf Strömen
  - Existenz kleinster Fixpunkte

- Starke Kausalität
  - Ausgabe zum Zeitpunkt t hängt nur von Eingaben vor Zeitpunkt t ab
  - Fixpunkte existieren und sind eindeutig
  - Nachweis durch induktive Konstruktion

Schnittstellenmodell für Systeme

$I = \{ x_1 , x_2 , ... \}$ Menge der Eingabekanäle

$O = \{ y_1 , y_2 , ... \}$ Menge der Ausgabekanäle

Syntaktische Schnittstelle: $(I \blacktriangleright O)$

Schnittstellenverhalten

$f : \vec{I} \rightarrow \vec{O}$

Zeitabhängige Kommunikationsgeschichten

Gegeben

$$z: C \rightarrow (IN_+ \rightarrow M^*)$$

wir definieren für $t \in IN$:

$$z{\downarrow}t$$

als den Zeitabschnitt (das Präfix) von x bis zum Zeitpunkt t:

$$z{\downarrow}t : C \rightarrow (\{1, ..., t\} \rightarrow M^*)$$

durch (für $i \in \{1, ..., t\}$)

$$(z{\downarrow}t)(c)(i) = z(c)(i)$$

Kausalität

f heißt stark kausal, falls für alle x, x′

$$x{\downarrow}t = x'{\downarrow}t \Rightarrow f(x){\downarrow}t+1 = f(x'){\downarrow}t+1$$

f heißt (schwach) kausal, falls für alle x, x′

$$x{\downarrow}t = x'{\downarrow}t \Rightarrow f(x){\downarrow}t = f(x'){\downarrow}t$$

Theorem zur Kausalität

Die Schnittstellenabstraktion

- einer Mealy Maschine ist schwach kausal
- einer Moore Maschine ist stark kausal

Endliche Approximation einer Funktion

Gegeben die stark kausale Funktion

$$f: (I \to (IN_+ \to M^*)) \to (O \to (IN_+ \to M^*))$$

Wir definieren für jedes $t \in IN$ eine Funktion

$$f^{(t)}: (I \to (\{1, ..., t\} \to M^*)) \to (O \to (\{1, ..., t+1\} \to M^*))$$

durch (mit $x: I \to (IN_+ \to M^*)$)

$$f^{(t)}(x{\downarrow}t) = f(x){\downarrow}t+1$$

Theorem:

(1) $f^{(t)}$ ist durch diese Gleichung eindeutig bestimmt

(2) gilt für $x: I \to (IN_+ \to M^*)$, $y: O \to (IN_+ \to M^*)$ für alle t:

$$f^{(t)}(x{\downarrow}t) = y{\downarrow}t+1$$

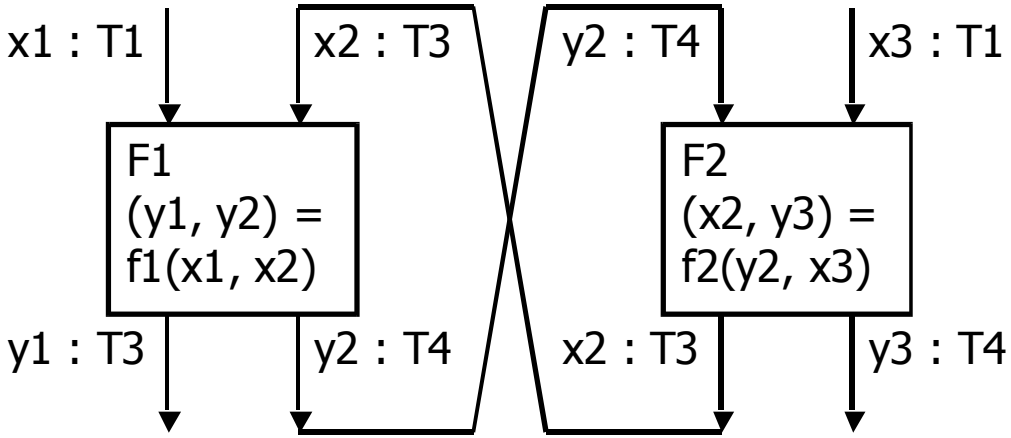
dann gilt $f(x) = y$

Gleichungssystem

Wir definieren für C = {x1, x2, x3, y1, y2, y3 } und gegebene Ströme s1, s3 die Funktion

$$g_{(s1, s3)}: (C \rightarrow (IN_+ \rightarrow IN)) \rightarrow (C \rightarrow (IN_+ \rightarrow IN))$$

durch



$g_{(s1, s3)}(z)(x1) = s1$

$g_{(s1, s3)}(z)(x3) = s3$

$(g_{(s1, s3)}(z)(y1), g_{(s1, s3)}(z)(y2)) = f1(z(x1), z(x2))$

$(g_{(s1, s3)}(z)(x2), g_{(s1, s3)}(z)(y3)) = f2(z(y2), z(x3))$

Konstruktion der Ströme rückgekoppelter Kanäle

Wir konstruieren eine Belegung aller Kanäle in C

$$z: C \rightarrow (IN_+ \rightarrow IN)$$

induktiv wie folgt: Wir definieren

$$z^{(t)} : (\{1, ..., t\} \rightarrow (IN_+ \rightarrow IN))$$

durch

$$z^{(0)}(c) = \langle\rangle$$
$$z^{(t+1)}(c) = g_{(s1, s3)}{}^{(t)}(z^{(t)})$$

Wir definieren für alle $t \in IN_+$:

$$z(c)(t) = z^{(t+1)}(c)(t)$$

Theorem: Es gilt

$$z = g_{(s1, s3)}(z)$$

Wir definieren für alle x:

$$(F1 \otimes F2)(x) = z$$

Komposition von Zustandsmaschinen

Gegeben Moore Maschinen $M_k = (\Delta_k, \Lambda_k)$ (für k = 1, 2):

$$\Delta_k: \Sigma_k \times (I_k \rightarrow M^*) \rightarrow \wp(\Sigma_k \times (O_k \rightarrow M^*))$$

Wir definieren die zusammengesetzte Zustandsmaschine $\Delta = \Delta_1 \parallel \Delta_2$

$$\Delta: \Sigma \times (I \rightarrow M^*) \rightarrow \wp(\Sigma \times (O \rightarrow M^*))$$

wie folgt

$$\Sigma = \Sigma_1 \times \Sigma_2$$

für $x \in I$ and $(\sigma_1, \sigma_2) \in \Sigma$ gelte:

$$\Delta((\sigma_1, \sigma_2), x) = \{((\sigma_1', \sigma_2'), z|O):\ x = z|I \wedge \forall\ k: (\sigma_k', z|O_k) \in \Delta_k(\sigma_k, z|I_k)\ \}$$

Wir schreiben

$$M = M_1 \parallel M_2 = (\Delta_1 \parallel \Delta_2\ ,\ \Lambda_1 \times \Lambda_2)$$

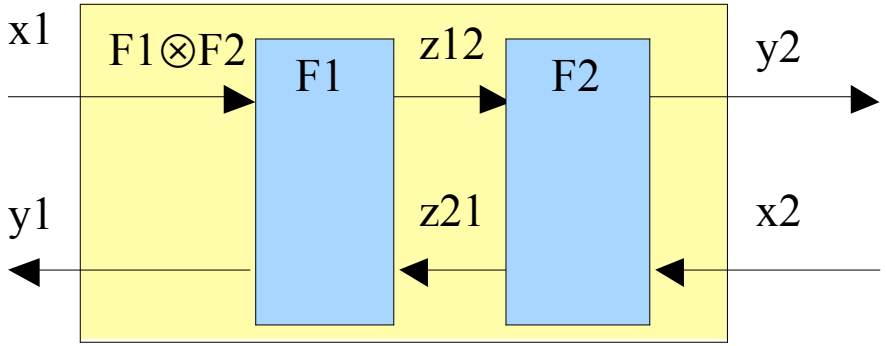
Eine wichtige Eigenschaft...

Schnittstellenabstraktion distributiert über Komposition

$$Abs((\Delta 1, \sigma 1) \,||\, (\Delta 2, \sigma 2)) =$$

$$Abs((\Delta 1, \sigma 1)) \otimes Abs((\Delta 2, \sigma 2))$$

Modulare Kompositionsregel für Spezifikationen



F1

in	x1, z21: T
out	y1, z12: T
P1	

F2

in	x2, z12: T
out	y2, z21: T
P2	

F1⊗F2

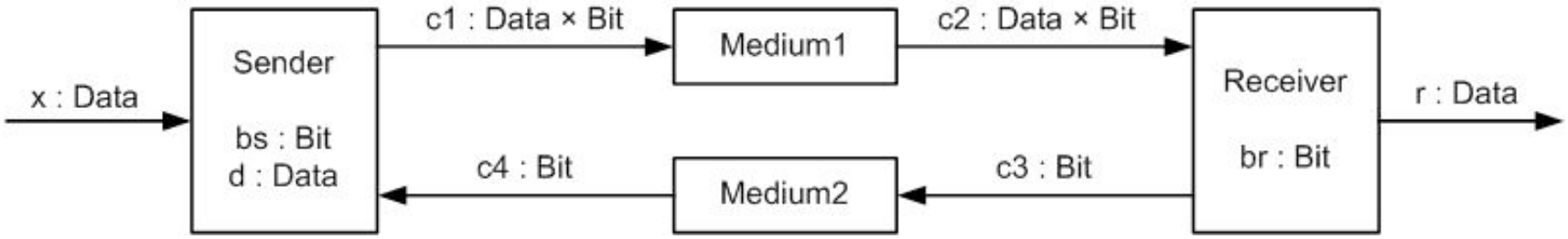
in	x1, x2: T
out	y1, y2: T
∃ z12, z21: P1 ∧ P2	

"Alternating-Bit"-Protokoll

Black Box Sicht auf das System

"Alternating-Bit"-Protokoll (1)

- Struktur- oder Verteilungssicht auf das System



Datenflussnetze

- Informationsfluss zwischen Komponenten modelliert durch
  - Kanalverbindungen
  - Datenströme
- Bildung von Datenflussnetzen durch parallele Komposition
  - Deklarationen von Strömen
  - Kanäle sind Identifikatoren für Ströme
- Beschreibung durch
  - Datenflussdiagramm
  - System rekursiver Gleichungen für Ströme auf den Kanälen

Parallele Komposition

O_1 und O_2 disjunkte Mengen von Kanälen

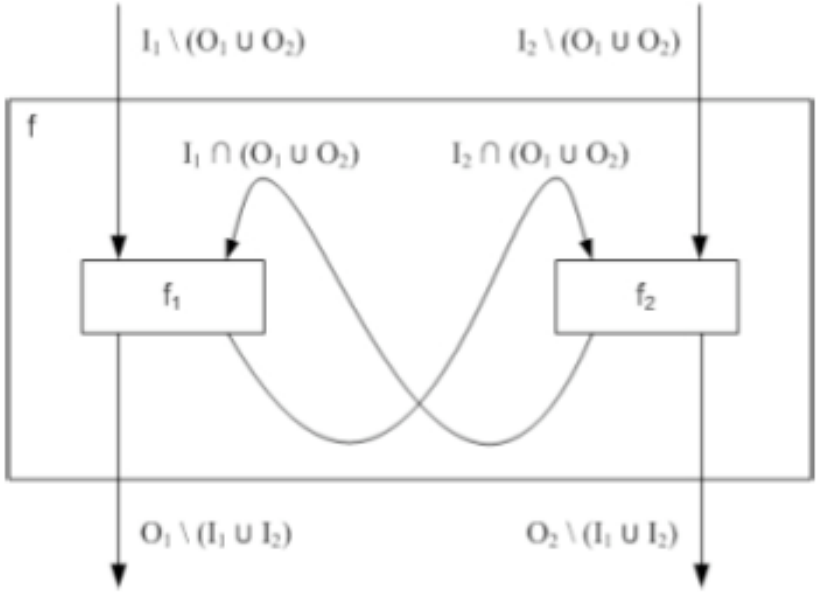
$f_k : \vec{I}_k \rightarrow \vec{O}_k$ mit $k = 1,2$ stromverarbeitende stark kasusale Funktionen

Für zusammengesetztes Systems gilt: $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$ und

$O = (O_1 \cup O_2) \setminus (I_1 \cup I_2)$

Verhalten gegeben durch Funktion

$f_1 \otimes f_2 : \vec{I} \rightarrow \vec{O}$

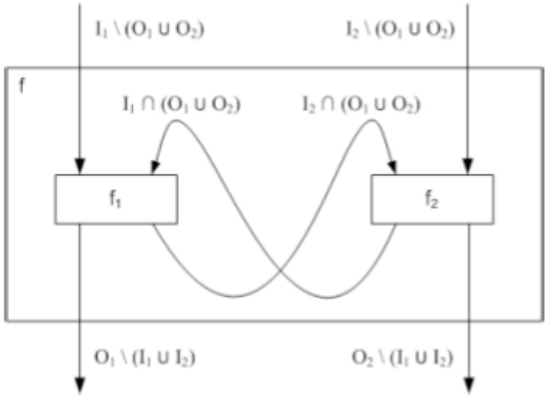


Parallele Komposition (2)

Für $x \in \vec{I}$ definieren wir

$(f_1 \otimes f_2)(x) = y$ wobei $y \in \vec{O}, y = z|_O$ und die Belegung $z \in \vec{C}$ der Kanäle in

$C = O_1 \cup O_2 \cup I_1 \cup I_2$



Fixpunkt der Gleichungen

$z|_{I_1 \cup I_2} = x$

$f_1(z|_{I_1}) = z|_{O_1}$

$f_2(z|_{I_1}) = z|_{O_2}$

$z|_{C'}$: Restriktion (Einschränkung) der Belegung z auf die Teilmenge $C' \subseteq C$.

Nichtdeterministische Komponenten

Darstellung durch Menge stromverarbeitender Funktionen:

$$F \subseteq \left\{ f : \vec{I} \rightarrow \vec{O} \right\}$$

Komposition der Mengen von Funktionen punktweise:

$$F_1 \otimes F_2 = \left\{ f_1 \otimes f_2 : f_1 \in F_1 \wedge f_2 \in F_2 \right\}$$

Kommunikationsnetze

- Modulare Darstellung stromverarbeitender Funktionen
  - Dekomposition in Kommunikationsnetze
  - Hierarchische Netze von Datenflussknoten

- Datenflussgraph
  - Eingangskanten: Kanten ohne Ursprung
  - Ausgangskanten: Kanten ohne Ziel
  - Jede Kante hat höchstens einen Ursprung, kann jedoch viele Ziele haben
  - Komponente
    - Wiederum durch ein Netz beschrieben *oder*
    - Verhalten durch stromverarbeitende Funktion beschrieben

Hierarchische Architekturen

Eine hierarchische Systemarchitektur besteht

Aus einem System

a) mit Schnittstellenspezifikation, und
b) mit Implementierung

Eine Implementierung ist gegeben durch

a) Eine Zustandsmaschine mit Ein/Ausgabekanälen, die die Schnittstellenspezifikation erfüllt (deren Schnittstellenabstraktion eine Funktion auf Strömen liefert, für die Spezifikation gilt), oder
b) Ein hierarchische Systemarchitektur, die die Schnittstellenspezifikation erfüllt (deren Schnittstellenabstraktion eine Funktion auf Strömen liefert, für die Spezifikation gilt),

Parallele Komposition ohne Rückkopplung

Seien

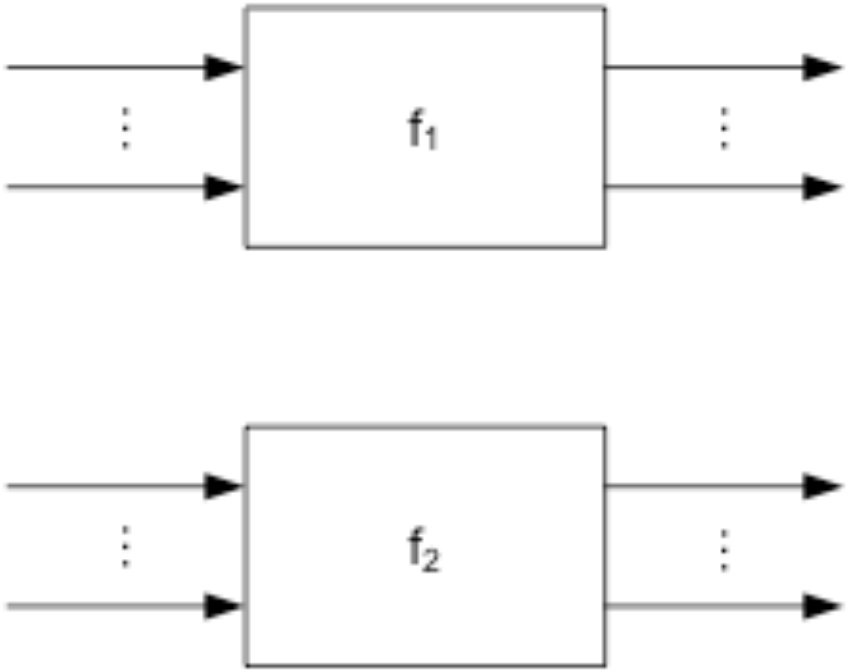
$f_1 : \vec{I}_1 \rightarrow \vec{O}_1$

$f_2 : \vec{I}_2 \rightarrow \vec{O}_2$

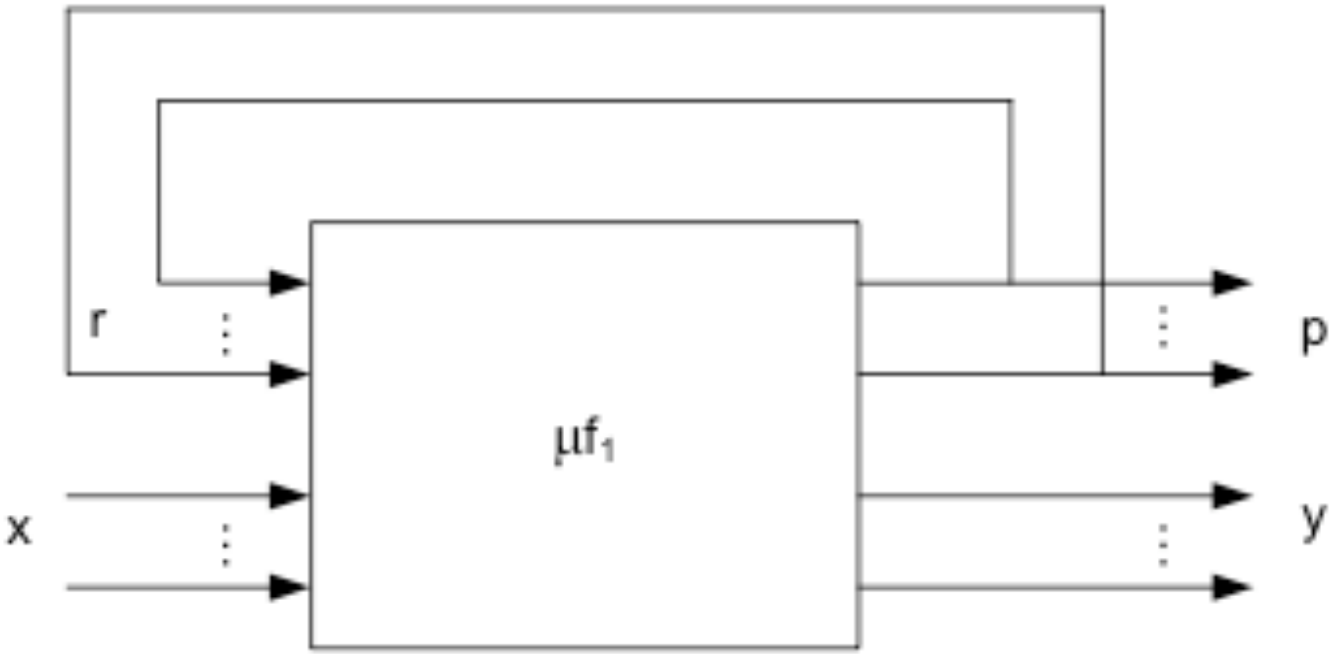
$O_1 \cap O_2 = \varnothing$

$O_1 \cap I_2 = \varnothing$

$I_1 \cap O_2 = \varnothing$



Rückkopplung

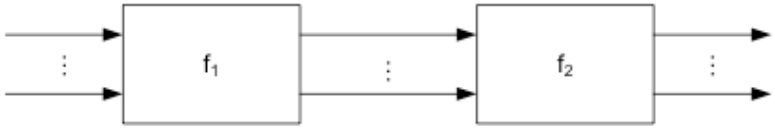


Sequentielle Komposition

Sei $O_1 = I_2$ dann ist $f_1 \; f_2$ eine stromverarbeitende Funktion mit Eingangskanälen I_1 und Ausgangskanälen O_2

Es gilt:

$(f_1 \; f_2)(x) = f_2(f_1(x))$



Kanäle in Anweisungen

Wir können auch die Sprache der bewachten Anweisungen

- um das Konzept der Kanäle erweitern
- um Anweisungen erweitern, die über Kanäle Sende- oder Empfangsaktionen vornehmen

Wir betrachten Programme der Form

$$[\![S_1 \,||\, S_2 \,||\, \ldots \,||\, S_n]\!]$$

über einer Menge C von typisierten Kanälen, wobei jeder Kanal $c \in C$

- in höchstens einer Anweisung S_i als Eingabekanal und
- in höchstens einer Anweisung S_k als Ausgabekanal

auftritt

Kanäle als gemeinsame Programmvariable

Ein Kanal c entspricht einer gemeinsamen Programmvariablen der Sorte *Sequence*, die am Anfang leer ist

Die Variable dient als Puffer (Warteschlange) und enthält die bisher in den Kanal geschrieben, aber nicht gelesenen Werte.

Sendeanweisung

Tritt c als Ausgabekanal auf, dann beschreibt die Anweisung

$$c\ !\ E$$

wobei E ein Ausdruck für einen Wert des Typs des Kanals ist,
das Senden des Wertes des Ausdrucks E über Kanal c.

$$\{Q[c^\circ\langle E\rangle/c]\}\ c\ !\ E\ \{Q\}$$

dabei steht die Sendeanweisung c ! E für die Zuweisung $c := c^\circ\langle E\rangle$

$$\{\ Q[c^\circ\langle E\rangle/c]\ \}\ c := c^\circ\langle E\rangle\ \ \{\ Q\ \}$$

Empfangsanweisung

Tritt c als Eingabekanal auf, dann beschreibt die Anweisung

$$c \,?\, v$$

wobei v eine Variable für einen Wert des Typs des Kanals ist,
das Senden des Wertes des Ausdrucks E über Kanal c.

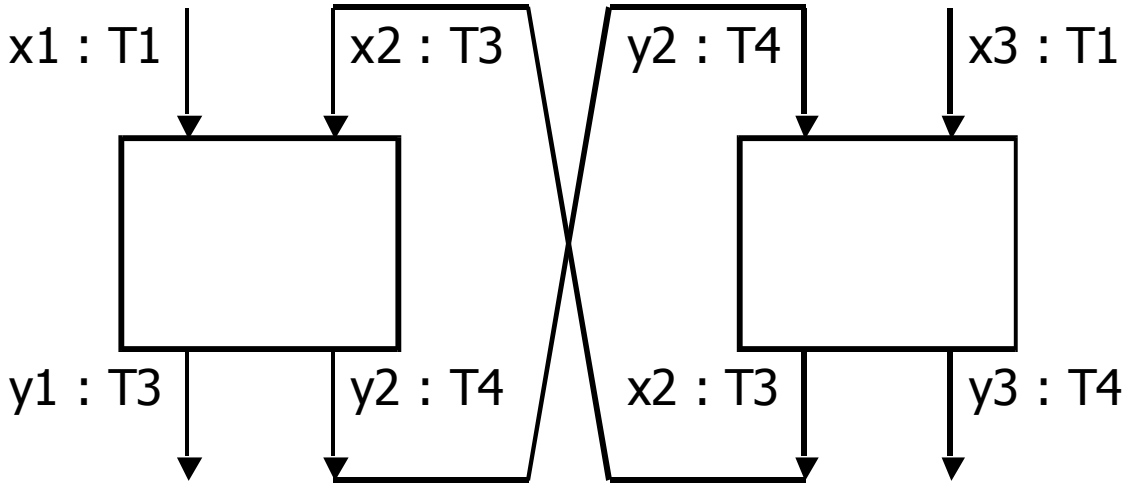
$$\{\, Q[first(c)/v, rest(c)/c] \vee c = \langle\,\rangle \,\}\ \ c \,?\, v\ \ \{Q\}$$

dabei steht die Sendeanweisung c ? E für die unteilbare Zuweisung
if $\langle$ c $\neq \langle\,\rangle$ then v, c := first(c), rest(c) $\rangle$ fi

$$\{\, Q[first(c)/v, rest(c)/c] \vee c = \langle\,\rangle \,\}$$
$$\text{if } \langle\ c \neq \langle\,\rangle \text{ then } v, c := first(c), rest(c) \,\rangle \text{ fi}$$
$$\{\, Q \,\}$$

System als Programm

```
[[
  do true then x1?v1; x2?v2; y1!v1+v2+1; y2!v1+v2+1 od
||
  x2!0; do true then y2?v3; x2!v3 od
||
  do true then x3?v4; y3!v4 od
]]
```



Abschließende Bemerkungen

- Systemarchitekturen auf Basis von Kanälen und Strömen beschreiben den Datenfluss zwischen den Komponenten
- Auch Algorithmen lassen sich so beschreiben
- Allerdings sollte in der Regel die Kommunikation zwischen den Komponenten einer Architektur eher sparsam sein
- Wir haben folgende unterschiedliche Konzepte behandelt
  - Systeme mit gezeiteten Strömen (zeitsynchrone Kommunikation) und starker Kausalität
  - Systeme mit ungezeiteten Strömen (asynchrone Kommunikation)