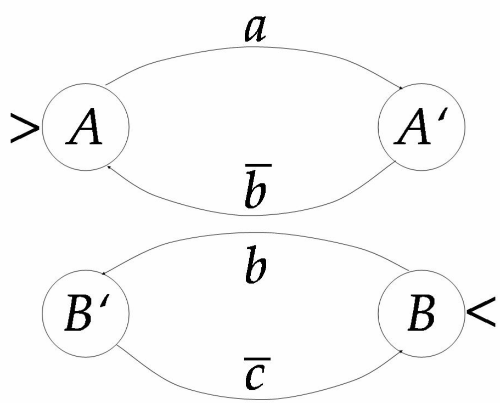
Milners Kalkül Kommunizierender Systeme (CCS)

Arnaud Fietzke
betreut durch Tim Priesnitz, Guido Tack

Proseminar: **Theorie kommunizierender Systeme**

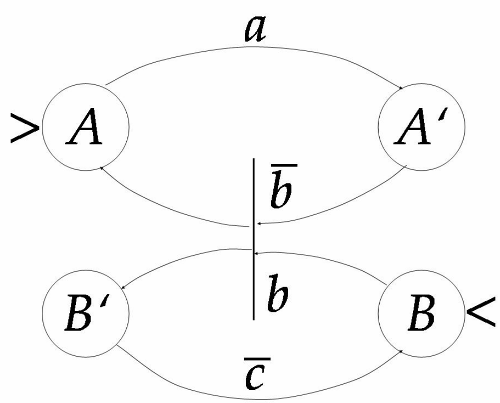
Programming Systems Lab – Prof. Gert Smolka

Modellierung nebenläufiger Systeme



$$A = a.A' \qquad B = b.B'$$
$$A' = \overline{b}.A \qquad B' = \overline{c}.B$$

Modellierung nebenläufiger Systeme



$A \mid B$ *mit*

$A = a.A'$ $\qquad B = b.B'$

$A' = \overline{b}.A$ $\qquad B' = \overline{c}.B$

CCS

Übersicht

- Einführung
- Syntax CCS
- Semantik CCS
  - Idee
  - Strukturelle Kongruenz
  - Reaktion

Einführung

Prozessalgebra:

- algebraische Modellierung von nebenläufigen Prozessen
- Darstellung komplexer Systeme mit Hilfe weniger Operatoren
- ermöglicht automatische Verifikation

Prozessalgebren: Ansätze

- **CCS** (Calculus of Communicating Systems)
[Milner '80]

- **CSP** (Communicating Sequential Processes)
[Hoare '85]

- **ACP** (Algebra of Communicating Processes)
[Bergstra & Klop '84]

- **LOTOS** (Language of Temporal Ordered Specification)
[Brinksma & Draft '88]

Syntax CCS

Definition:

Menge $\mathcal{P}$ der Prozessausdrücke:

$P ::= A\langle a_1,\ldots,a_n\rangle \mid \sum_{i\in I} \alpha_i.P_i \mid P_1 | P_2 \mid \text{new } a\ P$

mit I endliche Indexmenge

$\mathbf{0} := \sum_{i\in I} \alpha_i.P_i \quad \text{mit } I = \emptyset$

$\alpha \in \mathcal{N} \cup \overline{\mathcal{N}} \cup \{\ \tau\ \}$

$\mathcal{N} = \{a,b,c,\ldots\}$ Namen

System von nebenläufigen Prozessen:

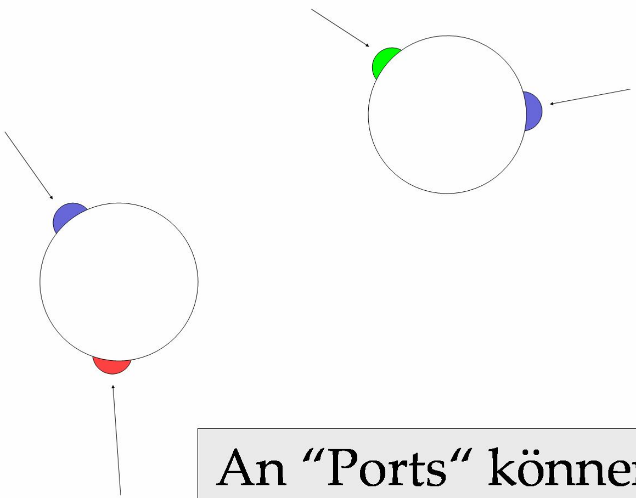
$P_1 \mid \ldots \mid P_n \quad$ mit

$$\left[\begin{array}{l} P_1 = \ldots \\ \vdots \\ P_n = \ldots \end{array}\right.$$

Semantik CCS

Idee: "chemical machine"

[Berry & Boudol '89]

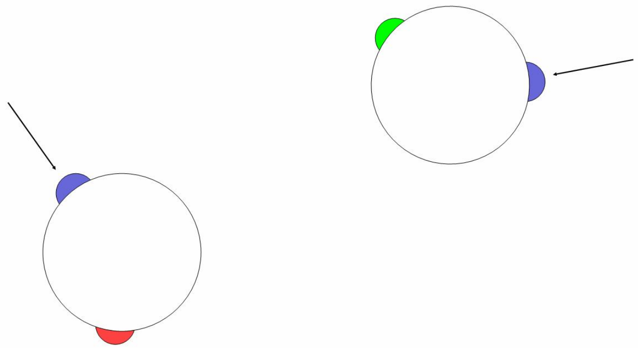


An “Ports“ können **beobachtbare Aktionen** stattfinden ($\alpha \in \boldsymbol{N} \cup \overline{\boldsymbol{N}}$)

Semantik CCS

Idee: "chemical machine"

[Berry & Boudol '89]

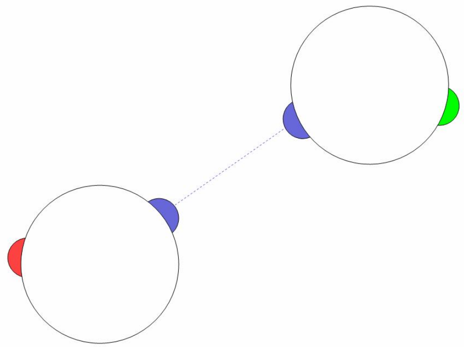


Komplementäre “Ports“ sind **Reaktion**spunkte
z.B. β und $\overline{\beta}$

Semantik CCS

Idee: "chemical machine"

[Berry & Boudol '89]



"Moleküle" können sich **annähern** und **reagieren**

Prozess-Kongruenz

Definition :

Äquivalenzrelation (reflexiv, symmetrisch, transitiv) $\cong$ über $\mathcal{P}$ mit

$$\left.\begin{array}{lcl} \alpha.P + M & \cong & \alpha.Q + M \\ \text{new } a\ P & \cong & \text{new } a\ Q \\ P \mid R & \cong & Q \mid R \\ R \mid P & \cong & R \mid Q \end{array}\right] \quad \text{falls } P \cong Q$$

Prozess-Kontext

Definition :

$\mathcal{C}$ ::= [] | $a.\mathcal{C}+M$ | new a $\mathcal{C}$ | $\mathcal{C} \mid P$ | $P \mid \mathcal{C}$

$\mathcal{C}[Q]$: Substitution von [] in $\mathcal{C}$ durch Q
Für $\mathcal{C}$=[] gilt: $\mathcal{C}[Q] = Q$

Eine bestimmte Prozess-Kongruenz $\cong$ lässt sich durch ein Gleichungssystem $\mathcal{E}$ definieren :

Prozess-Kongruenz

Eine bestimmte Prozess-Kongruenz $\cong$ lässt sich durch ein Gleichungssystem $\mathcal{E}$ definieren :

- $\cong$ erfüllt alle Gleichungen aus $\mathcal{E}$
- für jede Sequenz $Q_1,\ldots Q_n$ ($n \geq 1$) von Ausdrücken gilt $Q_1 \cong Q_n$ falls $Q_i = \mathcal{C}[P]$
  und $Q_{i+1} = \mathcal{C}[P']$
  und es gilt: $P \cong P'$ oder $P' \cong P$

Strukturelle Kongruenz

Definition:

Prozess-Kongruenz $\equiv$, definiert durch Gleichungen:

(1) Änderung gebundener Namen

Beispiel:
$(\text{new } b)\ a.b \ \equiv\ (\text{new } c)\ a.c$

Strukturelle Kongruenz

Definition:

Prozess-Kongruenz $\equiv$, definiert durch Gleichungen:

(1) Änderung gebundener Namen

(2) Umordnung der Terme in Summen

Beispiel:
$a.0 + b.0 \equiv b.0 + a.0$

Strukturelle Kongruenz

Definition:

Prozess-Kongruenz $\equiv$, definiert durch Gleichungen:

(1) Änderung gebundener Namen

(2) Umordnung der Terme in Summen

(3) $P \mid \mathbf{0} \equiv P, \quad P \mid Q \equiv Q \mid P, \quad P \mid (Q \mid R) \equiv (P \mid Q) \mid R$

Strukturelle Kongruenz

Definition:

Prozess-Kongruenz $\equiv$, definiert durch Gleichungen:

(1) Änderung gebundener Namen

(2) Umordnung der Terme in Summen

(3) $P \mid \mathbf{0} \equiv P$, $P \mid Q \equiv Q \mid P$, $P \mid (Q \mid R) \equiv (P \mid Q) \mid R$

(4) new $\boldsymbol{a}$ $(\boldsymbol{P} \mid \boldsymbol{Q}) \equiv \boldsymbol{P} \mid$ new $\boldsymbol{a}$ $\boldsymbol{Q}$ falls a nicht frei in P,
new $\boldsymbol{a}$ $\mathbf{0} \equiv \mathbf{0}$, new $\boldsymbol{ab}$ $\boldsymbol{P} \equiv$ new $\boldsymbol{ba}$ $\boldsymbol{P}$

Strukturelle Kongruenz

Definition:

Prozess-Kongruenz $\equiv$, definiert durch Gleichungen:

(1) Änderung gebundener Namen

(2) Umordnung der Terme in Summen

(3) $P \mid 0 \equiv P,\ P \mid Q \equiv Q \mid P,\ P \mid (Q \mid R) \equiv (P \mid Q) \mid R$

(4) new $a\ (P \mid Q) \equiv P \mid$ new $a\ Q$ falls a nicht frei in P,
new $a\ 0 \equiv 0$, new $ab\ P \equiv$ new $ba\ P$

(5) $\boldsymbol{A\langle\vec{b}\rangle \equiv \{\vec{b}/\vec{a}\}\, P_A}$ **falls** $\boldsymbol{A(\vec{a}) = P_A}$

Strukturelle Kongruenz

Definition:

Prozess-Kongruenz $\equiv$, definiert durch Gleichungen:

(1) Änderung gebundener Namen

(2) Umordnung der Terme in Summen

(3) $P \mid 0 \equiv P$, $P \mid Q \equiv Q \mid P$, $P \mid (Q \mid R) \equiv (P \mid Q) \mid R$

(4) … $a\ Q$ falls a nicht frei in P,
… $P \equiv$ new $ba\ P$

Notation:
$\vec{a}$ Sequenz von Namen $a_1,\ldots,a_n$

(5) $A\langle\vec{b}\rangle \equiv \{\vec{b}/\vec{a}\}\ P_A$ falls $A(\vec{a}) = P_A$

Standardform

Definition:

Ausdruck

$$\text{new } \overline{a}\ (M_1 \mid \ldots \mid M_n) \quad \text{mit } M_i \text{ nichtleere Summe } (1 \leq i \leq n)$$

ist in **Standardform**.

Falls n=0, $M_1 \mid \ldots \mid M_n = \mathbf{0}$
Falls $\overline{a}$ leer, fällt new $\overline{a}$ weg

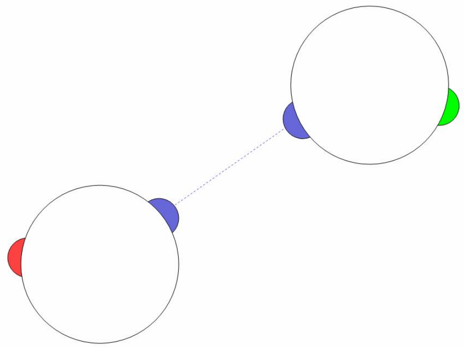
Theorem:

Jeder Prozessausdruck ist strukturell kongruent zu einer Standardform.

Semantik nebenläufiger Prozessausdrücke

Idee: "chemical machine"

[Berry & Boudol '89]

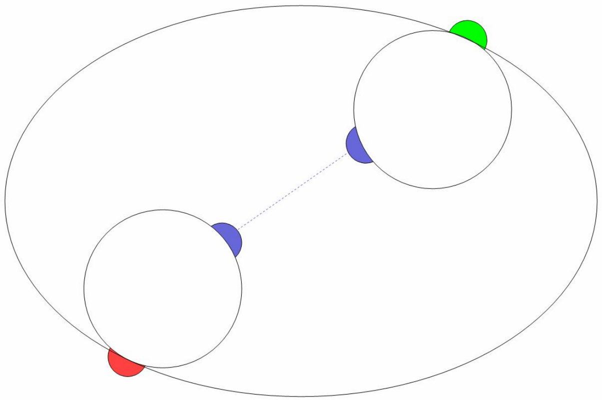


"Moleküle" können sich **annähern** und **reagieren**

Semantik nebenläufiger Prozessausdrücke

Idee: "chemical machine"

[Berry & Boudol '89]

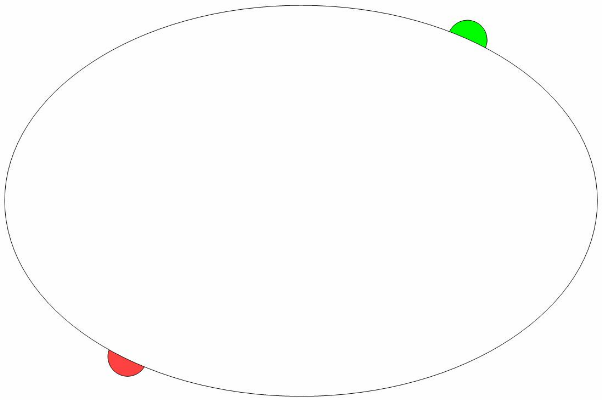


"Moleküle" können sich **annähern** und **reagieren**

Semantik nebenläufiger Prozessausdrücke

Idee: "chemical machine"

[Berry & Boudol '89]



"Moleküle" können sich **annähern** und **reagieren**

Reaktionen sind von aussen nicht mehr beobachtbar:

τ-Transitionen

Reaktion

Definition:

Relation $\rightarrow$ auf $\mathcal{P}$ wird durch Regeln definiert:

$$\text{TAU}: \quad \tau.P + M \rightarrow P$$

$$\text{REACT}: (a.P + M) \mid (\bar{a}.Q + N) \rightarrow P \mid Q$$

$$\text{PAR}: \quad \frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q}$$

$$\text{RES}: \quad \frac{P \rightarrow P'}{\text{new } a\, P \rightarrow \text{new } a\, P'}$$

$$\text{STRUCT}: \quad \frac{P \rightarrow P'}{Q \rightarrow Q'} \quad \text{falls } P \equiv Q \text{ und } P' \equiv Q'$$

Beispiel

Inferenzbaum:

$A' \mid B$ mit

$A = a.A'$ $\qquad$ $B = b.B'$

$A' = \overline{b}.A$ $\qquad$ $B' = \overline{c}.B$

$$\dfrac{\dfrac{}{\overline{b}.A \mid b.B' \rightarrow A \mid B'}\ \text{REACT}}{\boldsymbol{A' \mid B \rightarrow A \mid B'}}\ \text{STRUCT}$$

Beispiel

Alternative Reaktionen:

$$P \;=\; a.0 \mid \bar{a}.A \mid \bar{a}.B$$

Zwei Reaktionen sind möglich:

$$P \rightarrow A \mid \bar{a}.B$$

Beispiel

Alternative Reaktionen:

$$P = a.0 \mid \bar{a}.A \mid \bar{a}.B$$

Zwei Reaktionen sind möglich:

$$P \to A \mid \bar{a}.B$$

und $P \to \bar{a}.A \mid B$

Indeterminismus durch Reaktion

Referenzen

- Milner, R., Communicating and Mobile Systems: the π – Calculus, Cambridge University Press, 1999

- Milner, Operational & Algebraic Semantics of Concurrent Processes
  Handbook of Theoretical Computer Science B, Elsevier, 1990