Der π-Kalkül [Milner, Parrow, Walker]

- "The π-calculus is a model of concurrent computation based upon the notion of *naming*."
- Kompakt, klein, flexibel **aber** spartanisch, umständlich.
- Basismodell für theoretische Betrachtungen über Nebenläufigkeit— keine Spezifikations- oder Implementationssprache.

  - λ-Kalkül $\leftrightarrow$ π-Kalkül
    Haskell, ML $\leftrightarrow$ CSP

Der π-Kalkül: Syntax

Gegeben: unendliche Menge $\mathcal{X}$ von Variablen.

Defn: *Prozesse:* $P, Q \in \mathcal{P}$

$$P ::= P \mid Q \;\mid\; !P \;\mid\; (\nu x)P$$

Defn: Prozesse in *Normalform:* $N, M \in \mathcal{N}$

$$N ::= \pi.P \;\mid\; \mathbf{0} \;\mid\; M + N$$

Zwei unterschiedliche Formen von Präfixen:

- $x(y)$ — bindet y, “lies Namen y über Kanal x”
- $\overline{x}y$ — bindet y *nicht*, “schreibe Namen y auf Kanal x”

Der π-Kalkül: Erste Beispiele

- Notation: $x(y)$ für $x(y).\mathbf{0}$
- Aus $\overline{x}y \mid x(u).\overline{u}v \mid \overline{x}z$ wird $\mathbf{0} \mid \overline{y}v \mid \overline{x}z$ oder $\overline{x}y \mid \overline{y}z \mid \mathbf{0}$
- Aus $(\nu x)(\overline{x}y \mid x(u).\overline{u}v) \mid \overline{x}z$ kann nur $\mathbf{0} \mid \overline{y}v \mid \overline{x}z$ werden
- $\overline{x}y \mid !x(u).\overline{u}v \mid \overline{x}z$ kann zu $\mathbf{0} \mid \overline{y}v \mid !x(u).\overline{u}v \mid \overline{x}z$ werden, daraus kann $\mathbf{0} \mid \overline{y}v \mid !x(u).\overline{u}v \mid \overline{y}z \mid \mathbf{0}$ werden

Der π-Kalkül: Operationelle Semantik I

Defn: fn(P), *freie* Namen in P

Defn: bn(P), *gebundene* Namen in P

Defn: *Strukturelle Kongruenz* $\equiv$ kleinste Kongruenzrelation auf $\mathcal{P}$:

(1) α-Kongruenz: gebundene Variablen gleich.

(2) $(\mathcal{N}/\equiv, +, \mathbf{0})$ ist ein symetrischer Monoid

(3) $(\mathcal{P}/\equiv, |, \mathbf{0})$ ist ein symetrischer Monoid

(4) $!P \equiv P \mid !P$

(5) $(\nu x)\mathbf{0} \equiv \mathbf{0}, (\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$

(6) $x \notin \mathrm{fn}(P) \Rightarrow (\nu x)(P \mid Q) \equiv P \mid (\nu x)Q$

Der π-Kalkül: Operationelle Semantik II

$$\text{Comm}\quad \frac{}{(R_1 + x(y).P) \mid (R_2 + \overline{x}z.Q) \to P[z/y]}$$

$$\text{Par}\quad \frac{P \to P'}{P \mid Q \to P' \mid Q}$$

$$\text{Res}\quad \frac{P \to P'}{(\nu x)P \to (\nu x)P'}$$

$$\text{Struct}\quad \frac{Q \equiv P \quad P \to P' \quad P' \equiv Q'}{Q \to Q'}$$

Der π-Kalkül: Rekursive Definitionen

Rekursive Definitionen durch Replikation:

Lösung für eine rekursive Gleichung

$$A(x) \stackrel{def}{=} P$$

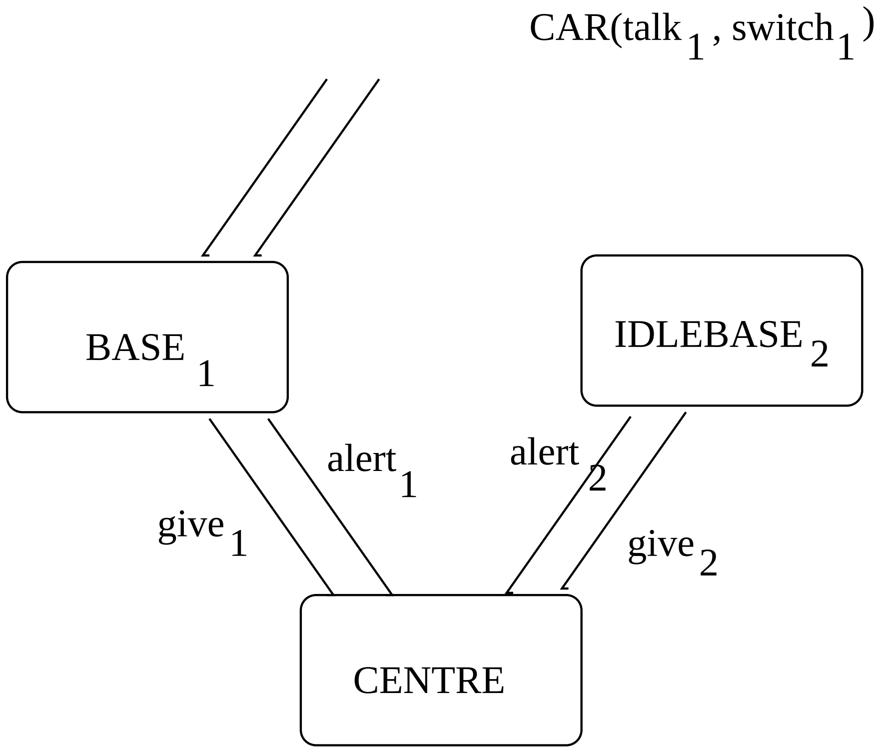
gegeben durch

$$!a(x).\hat{P},$$

wobei $\hat{P} = P[\overline{a}y/A(y)$. Beim Aufruf $A(z)$ ersetzen durch

$$(\nu a)(\overline{a}z \mid !a(x).\hat{P}.$$

Der π-Kalkül, Beispiel: das Autotelephon



Der π-Kalkül, Beispiel: das Autotelephon

$$
\begin{aligned}
\mathrm{CAR}(\mathit{talk}, \mathit{switch}) &\overset{\mathit{def}}{=} \mathit{talk}.\mathrm{CAR}(\mathit{talk}, \mathit{switch}) \\
&\quad + \mathit{switch}(\mathit{talk}', \mathit{switch}').\mathrm{CAR}(\mathit{talk}', \mathit{switch}') \\[1em]
\mathrm{BASE}(t, s, g, a) &\overset{\mathit{def}}{=} t.\mathrm{BASE}(t, s, g, a) \\
&\quad + g(t's').\overline{s}t's'.\mathrm{IDLEBASE}(t, s, g, a) \\[1em]
\mathrm{IDLEBASE}(t, s, g, a) &\overset{\mathit{def}}{=} a.\mathrm{BASE}(t, s, g, a) \\[1em]
\mathrm{BASE}_i &\overset{\mathit{def}}{=} \mathrm{BASE}(\mathit{talk}_i, \mathit{switch}_i, \mathit{give}_i, \mathit{alert}_i) \quad i = 1, 2 \\[1em]
\mathrm{IDLEBASE}_i &\overset{\mathit{def}}{=} \mathrm{IDLEBASE}(\mathit{talk}_i, \mathit{switch}_i, \mathit{give}_i, \mathit{alert}_i) \quad i = 1, 2
\end{aligned}
$$

Der π-Kalkül, Beispiel: das Autotelephon

$$\mathrm{CENTRE}_1 \stackrel{\mathit{def}}{=} \overline{\mathit{give}_1}\,\mathit{talk}_2\mathit{switch}_2.\mathit{alert}_2.\mathrm{CENTRE}_2$$

$$\mathrm{CENTRE}_2 \stackrel{\mathit{def}}{=} \overline{\mathit{give}_2}\,\mathit{talk}_1\mathit{switch}_1.\mathit{alert}_1.\mathrm{CENTRE}_1$$

$$\begin{aligned}\mathrm{SYSTEM} \stackrel{\mathit{def}}{=}\; & (\nu\,\mathit{talk}_1, \mathit{talk}_2, \mathit{switch}_1, \mathit{switch}_2, \mathit{give}_1, \mathit{give}_2, \mathit{alert}_1, \mathit{alert}_2) \\ & (\mathrm{CAR}(\mathit{talk}_1, \mathit{switch}_1) \mid \mathrm{BASE}_1 \mid \mathrm{IDLEBASE}_2 \mid \mathrm{CENTRE}_1)\end{aligned}$$

Mögliche Verfeinerungen: mehr Kanäle, mehr Basen, kein festes $(\mathit{talk}, \mathit{switch})$-Kanalpaar pro Basis ...

Der π-Kalkül: Erweiterungen

- Vorgestellt: *monadischer* π-Kalkül
- Erweiterung: der *polyadische* π-Kalkül mit *Abstraktionen* und *Konkretisierungen* und mit Sorten.
- π-Kalkül höherer Ordnung
- Operationale Semantik durch *commitments*, Bisimulation; Axiomatisierung (endliche Prozesse, ohne Replikation).