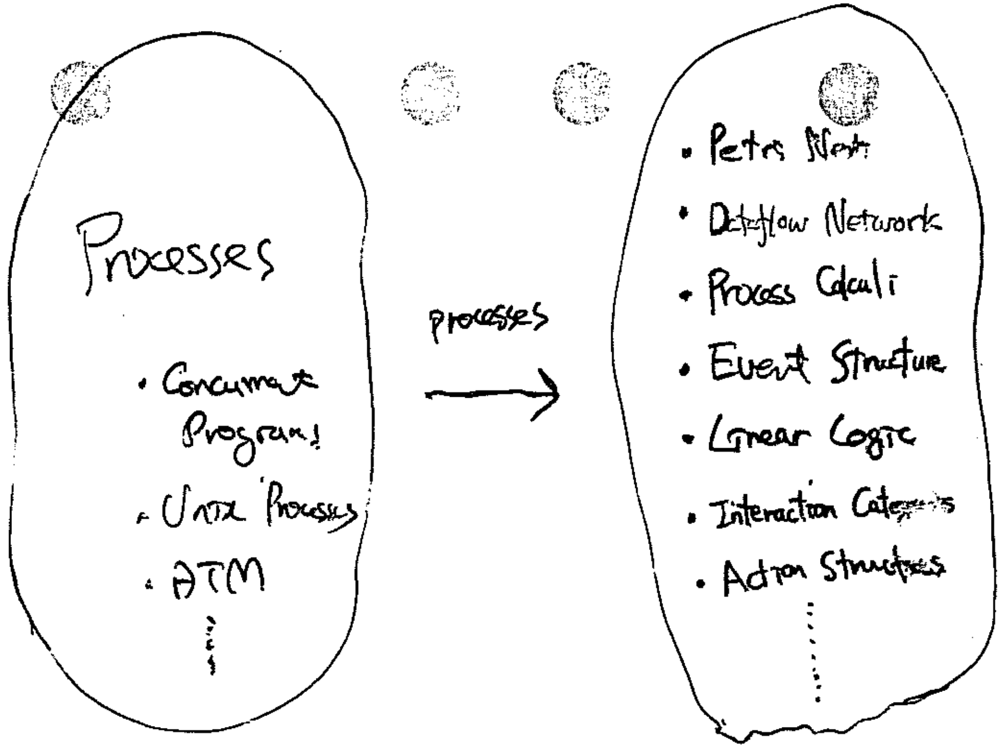
Processes — An Elementary Approach

Kohei Honda

Edinburgh University

Modelling Processes.



Starting Point.

π-Calculus. ([MPW89] → [Milner90])

$$P ::= ax.P \mid \bar{a}v.Q \mid P|Q \mid (\nu a)P \mid !ax.P \mid 0$$

(via asynchronous calculus)

$\bar{a}b.P \Rightarrow \bar{a}b.$

CC ([HY94a, HY94b])

$$P ::= \varsigma^{(n)}(a_1 \cdots a_n) \mid (\nu a)P \mid P|Q \mid 0$$

$\{\varsigma_1^{(n)}, \varsigma_2^{(m)}, \ldots\}$: a finite set of atoms

Process and Names.

- What are names for?
  - variables, identifiers, ...
- No structural difference between:

  $a.b.0$ and $e.f.0$

- Indeed:

  structural correspondence

  $a.b + b.a \sim a|b$

  $e.f + f.e \sim f|e$ etc.

Rooted Process Structure (1)

Definition.

Fix $\mathcal{N}$, a set of names $(a, b, c, \ldots)$.
Then a rooted process structure is given by:

- $\mathcal{P}$, a set of processes. $(P, Q, R, \ldots)$
- $FN : \mathcal{P} \to 2^{\mathcal{N}}$ the free name function.
- $[\sigma] : \mathcal{P} \to \mathcal{P}$ for each bijection σ on $\mathcal{N}$ s.t.

  (i) $P[\sigma_2 \circ \sigma_1] = P[\sigma_1][\sigma_2]$

  (ii) $\forall a \in FN(P).\ \sigma(a) = a \Rightarrow P[\sigma] = P$.

  (iii) $FN(P[\sigma]) = \sigma(FN(P))$.

Rooted Process Structure (2)

Examples of Rooted Process Structures

(1) Any process calculi we know, possibly modulo structural equality.

- $P|Q \equiv Q|P \quad (P|Q)|R \equiv P|(Q|R) \ \ldots$

  Then: $[P]_{\equiv}$ is a Rooted Process.

(2) Any (parallel) programming languages we know.

- Variables = names.

(3) λ-calculus.

Algebra of RPS (2)

Prop.

Let $\sim$ be a congruence on $\mathcal{P}$.

Define $\mathcal{P}/\sim$:

(1) Processes : $\{[P]_\sim\}$

(2) Free Names : $FN([P]_\sim) \overset{\text{def}}{=} \bigcap_{P' \sim P} \{FN(P')\}$

(3) Renaming : $[P]_\sim[\sigma] \overset{\text{def}}{=} [P[\sigma]]_\sim$.

Then $\mathcal{P}/\sim$ is an RPS.

Algebra of RPS (3)

Prop.

Let $\mathcal{P}$ and $\mathcal{Q}$ be process structures.

Then define $\mathcal{P} \times \mathcal{Q}$ by :

(1) Processes : $\{\langle P, Q\rangle \mid P \in \mathcal{P},\ Q \in \mathcal{Q}\}$

(2) Free names : $FN(\langle P, Q\rangle) \overset{\text{def}}{=} FN(P) \cup FN(Q)$.

(3) Renaming : $\langle P, Q\rangle[\sigma] \overset{\text{def}}{=} \langle P[\sigma], Q[\sigma]\rangle$.

Then $\mathcal{P} \times \mathcal{Q}$ is a process structure, with the usual universality property.

RPS and RPSrel.

- RPS (functional universe)

  objects : Process structures.

  arrows : Homomorphisms.

- RPSrel. (relational universe)

  objects : Process structure.

  arrows : Compatible Relations equipped with :

  $\cap$, $\cup$, $(\cdot)^{\circ}$, $-$, $(\ )^{c}$

Passage to Pinapples

$a(z).\,(x.\bar{z}c.\,c(x_1).$
$c(x_2).\,P$ ....

$\bar{a}b.P$

$a(z).\bar{z}v_1 + a'(z).\bar{z}'v_2 +$

....

$\Rightarrow$

Symmetry. (1).

<u>Definition.</u>

A symmetry σ of a process $P \in \mathcal{P}$ is a permutation over $FN(P)$ such that

$P[\sigma] = P$.

$D(abc) \sim D(acb)$

$D(abc) \stackrel{def}{=} a(x).(\overline{b}x \mid \overline{c}x)$

<u>Prop.</u>

The set of symmetries of P, written $Sym(P)$ forms a permutation group.

cf. isotropy, stabilizer.

Symmetry (2).

- The symmetry-representation of

$P,\ P[\sigma],\ P[\sigma'], \ldots\ldots$	$Sym(P)$
$Q,\ Q[\sigma],\ Q[\sigma'], \ldots$	$Sym(Q)$
$R,\ R[\sigma],\ R[\sigma'], \ldots$	$Sym(R)$
$\vdots$	$\vdots$

* Each orbit (consisting of <u>INFINITE</u> processes) is represented by <u>ONE</u> $Sym(P)$

Symmetry (3)

Prop (symmetries).

P_1 and P_2 are isomorphic iff there is a bijection $\mathcal{H}$ between their symmetry representations such that:

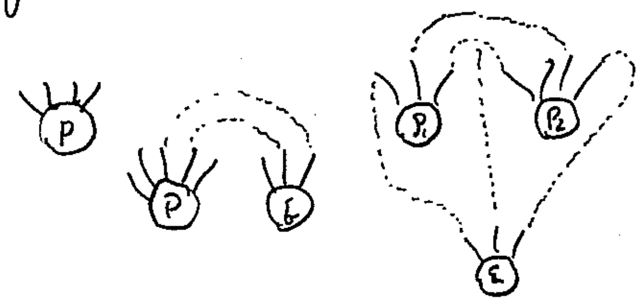
$$\mathcal{H}(Sym(P_1)) = Sym(P_2)$$

$$\Rightarrow \exists \sigma.\quad Sym(P_1) = \sigma \cdot Sym(P_2) \cdot \sigma^{-1}$$

bijection from $FN(P_1)$ to $FN(P_2)$.

$P\ P[\sigma]\ P[\sigma']$ ---	s	t	$P'\ P'[\sigma]$ ---
$Q\ Q[\sigma]\ Q[\sigma']$ ---	s'	t'	$Q'\ Q'[\sigma]$ ---
$R\ R[\sigma]\ R[\sigma']$ ---	s''	t''	$R'\ R'[\sigma]$ ---
⋮			⋮

Theory of Nameless Processes and Equivalence Theorem



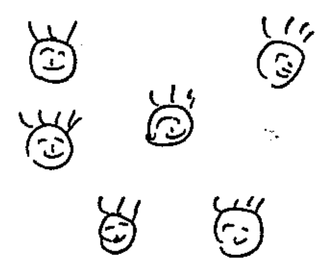
$a x.P \quad a v.\bar{z}v.Q$

$\bar{a}b + \tau e$ $\Leftarrow$

$P|Q$ $\Rightarrow$

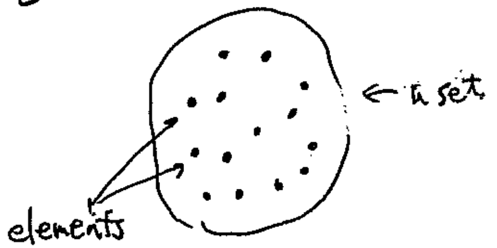
$\bar{a}z.\bar{z}v.\bar{z}u$

$az(\bar{z}a|\bar{z}v)$

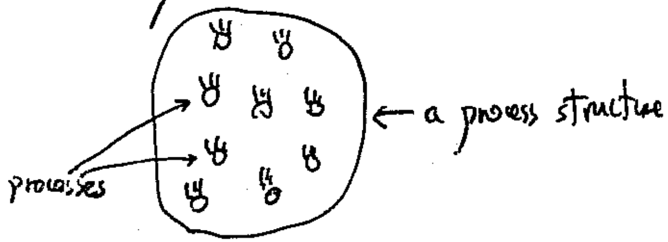


What is a Process Structure?

* Set theory provides a way to manipulate elements collectively.



* Theory of Process Structure offers a way to manipulate processes collectively.



Process Structure (1)

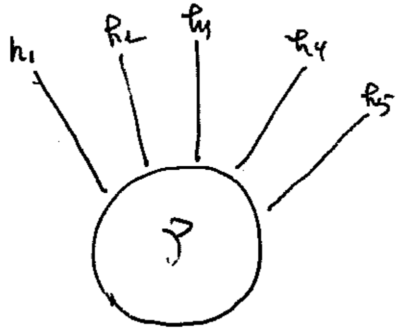
<u>Def.</u>

A process structure $\mathbf{P}$ is given by:

(i) $\mathbf{P}$: A set of <u>pure processes</u> $(p, q, r, \ldots)$

(ii) $\mathcal{H}(p)$: <u>Handles</u> of p; finite.

(iii) $S(p)$: <u>Symmetries</u> of p, a permutation group over $\mathcal{H}(p)$.



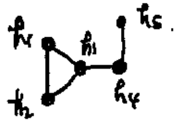
$$\left\{ \begin{pmatrix} h_1 & h_2 \\ h_2 & h_1 \end{pmatrix}, \; id_{\mathcal{H}(p)} \right\}$$

Process Structure (3)

* Examples of process structure?

(1) Any set. (P) = an element.

(2) Dataflow, Proof Net, π net. 

(3) A set of graphs. 

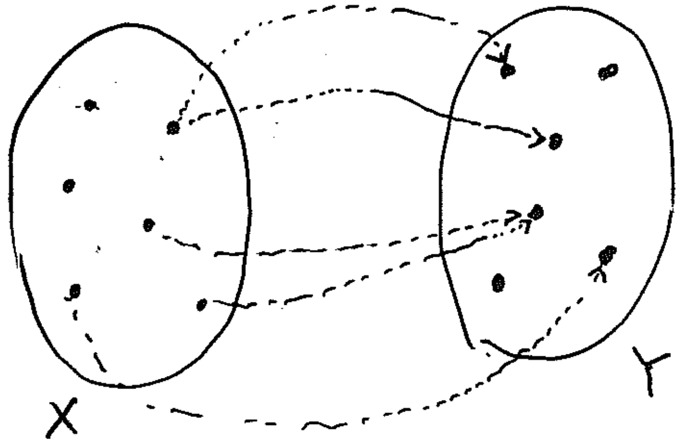
(4) A set of arrows on a category.

(5) An indexed family of permutation groups (Definition!?).

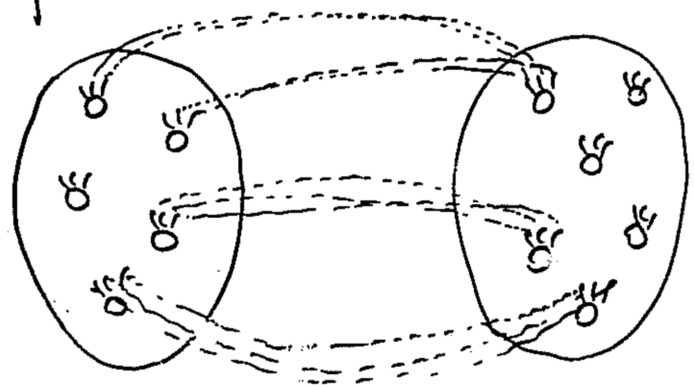
(6) Any symmetry presentation.

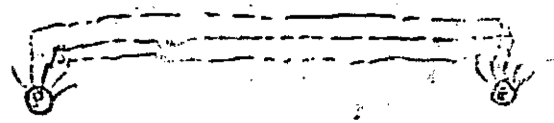
Relating Processes.

• In sets we have:



• For processes we get:

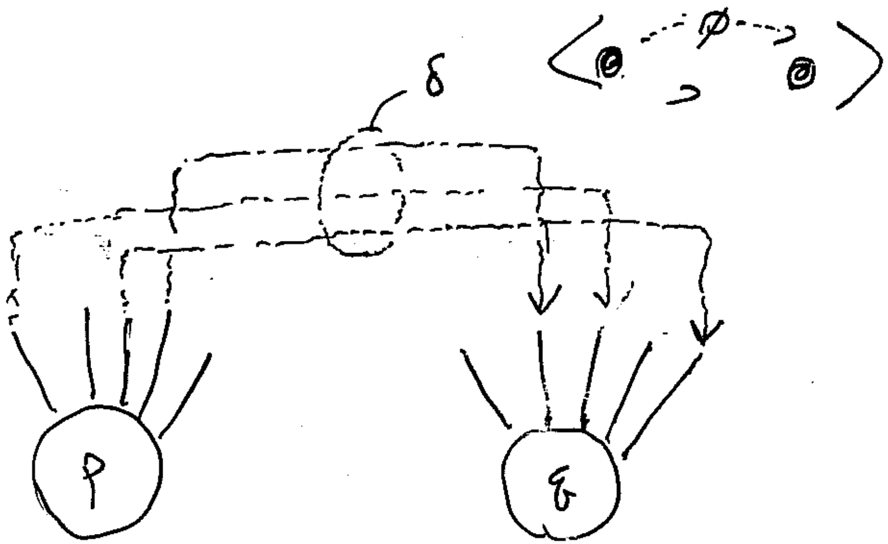




Correspondence (1)

Def.

A correspondence from p to q is a triple $\langle p, \delta, q\rangle$ where δ is a partial injection from $\mathcal{H}(p)$ to $\mathcal{H}(q)$.



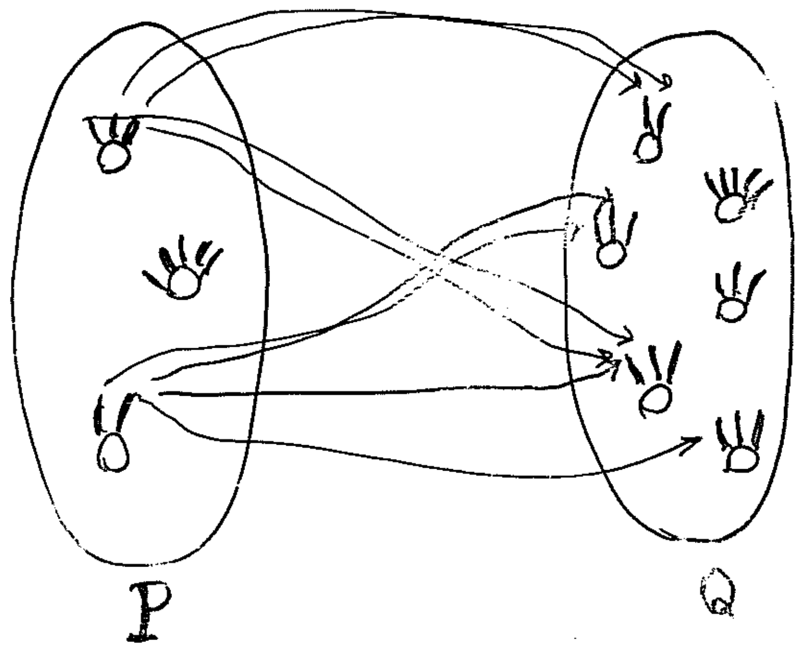
$*$ $\langle p, \delta, q\rangle^{-1} = \langle q, \delta^{-1}, p\rangle$

P-relation and P-map (1)

Def.

Given $\mathbf{P}$ and $\mathbf{Q}$, a p-relation $\mathcal{R}$ is a set of correspondences from processes in $\mathbf{P}$ to processes in $\mathbf{Q}$ s.t.

$$\langle p, \delta, q\rangle \in \mathcal{R} \text{ and } \delta' \sim \delta \Rightarrow \langle p, \delta', q\rangle \in \mathcal{R}.$$



P-relation and P-map (2)

Def

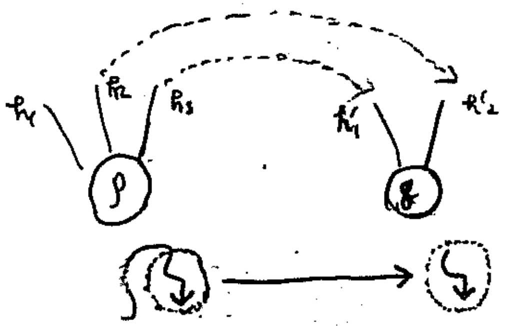
A prelation $R : P \to Q$ is a p-map iff:

(1) (totality and uniqueness) For each $p \in P$, there is a unique $\langle p, \delta, \varepsilon \rangle \in R$ up to $\sim$.

(2) (ind. surjectivity) $\langle p, \delta, \varepsilon \rangle \in R \Rightarrow \delta$ surjective.

(3) (symmetry preservation) If $\langle p, \delta, \varepsilon \rangle \in R$ then:

$$\forall \sigma \in S(p).\ \exists \sigma' \in S(\varepsilon).\ \ \delta \circ \sigma = \sigma' \circ \delta.$$



$$\begin{tikzcd} p \arrow[r, "\delta"] \arrow[d, "\sigma"'] & \varepsilon \arrow[d, "\sigma'"] \\ p \arrow[r, "\delta"'] & \varepsilon \end{tikzcd}$$

P-relation and P-map (3)

Prop.

If $F_1 : P \to Q$ and $F_2 : Q \to R$ is a p-map, then $F_2 \circ F_1$ is again a p-map.

Proof: For uniqueness, assume

$$\langle p, \delta_1, \varepsilon \rangle \in F_1, \quad \langle \varepsilon, \delta_2, r \rangle \in F_2.$$

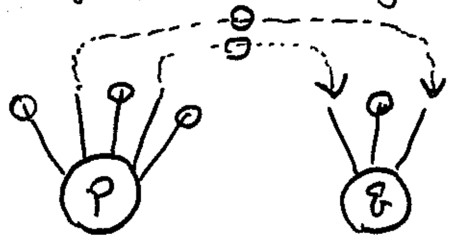
Then:

$$\sigma_3' \circ \delta_2 \circ \sigma_2 \circ \delta_1 \circ \sigma_1 = \sigma_3 \circ \sigma_3' \circ \delta_2 \circ \delta_1 \circ \sigma_1$$

$$\sim \delta_2 \circ \delta_1 \qquad \square$$

Product.

* A correspondence as a process:

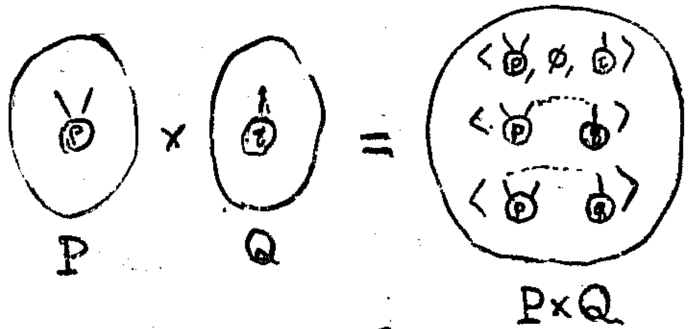


with: $\delta(\langle p, \delta, \varepsilon\rangle) \stackrel{def}{=} \{\langle \sigma_1, \sigma_2\rangle \mid \delta = \sigma_2 \circ \delta \circ \sigma_1\}$

* $P \times Q$: all correspondences from P to Q.

  - Usual universality.

  - Note:

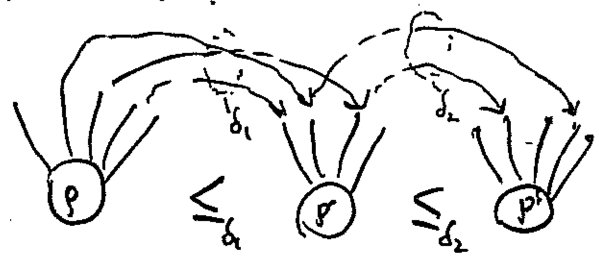


$1 \times 1 = 3.$

Pre-order, Equivalence, Quotient (1)

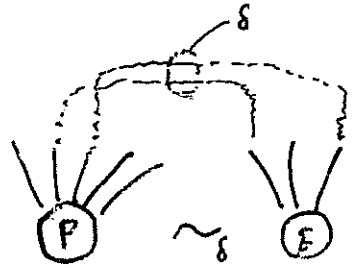
* A pre-order R over P means:

$$R \supseteq ID_P$$
$$R \supseteq R \circ R$$



* An equivalence R over P means:

$$R \supseteq ID_P$$
$$R \supseteq R \circ R$$
$$R^{-1} = R$$



$\bar{a} \sim (c)(\bar{a} \mid c.b)$

Pre-order, Equivalence, and Quotient (2)

Def.

A quotient of P by an equivalence $\sim$, written $P/\sim$, is given by:

(i) Processes: $\{[p]_\sim \mid p \in P\}$ Select $p' \in [p]_\sim$ for each equivalence class.

(ii) Handles: $\mathcal{H}([p]_\sim) \overset{\text{def}}{=} \mathcal{H}(p)[\sim]$

i.e. $\bigcap \{\delta(\mathcal{H}(p)) \mid q \sim_\delta p\}$

(iii) Symmetries: $\mathcal{S}([p]_\sim) \overset{\text{def}}{=} \{\theta \restriction \mathcal{H}([p]_\sim) \mid p \sim_\theta p\}$

Prop.

$P/\sim$ is a process structure. Moreover different choices of representatives result in isomorphic structures.

PS and PSrel (2)

- $\mathbf{PS}^{+}_{rel}$

objects: as in $\mathbf{PS}_{rel}$.

arrows: as in $\mathbf{PS}_{rel}$.

composition:

$$R_1 ; R_2 = \{\delta \geq \delta_1 ; \delta_2 \mid \delta_1 \in R_1 \land \delta_2 \in R_2\}.$$

Main Result

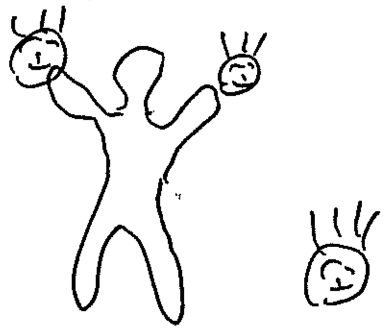
<u>Theorem</u>

PS and RPS are categorically equivalent.

Remark: PS_{rel} and RPS_{rel} are not equivalent. But PS^{+}_{rel} and RPS_{rel} are.

How to use Pineapples.

— Application to Types for Concurrency —



PS and PSrel. (1)

- PS.

  Objects: Process structures.

  Arrows: p-omaps.

  - ID_P: $\{\langle p, \sigma, p\rangle \mid p \in P,\ \sigma \in S(p)\}$.
  - Isomorphisms: A p-omap F s.t. F^{-1} is also a p-omap.

  ⋮

- PSrel

  objects: Process structures.

  Arrows: p-relation with operations:

  $$\cap \quad \cup \quad (\cdot)^{o} \quad - \quad (\,)^{c}$$

  - ID and iso's are as PS.

Basic Idea (1).

- Composition of functions:

  $\lambda x.\, x+1$ ($NAT \to NAT$) • 3 (NAT) ◎

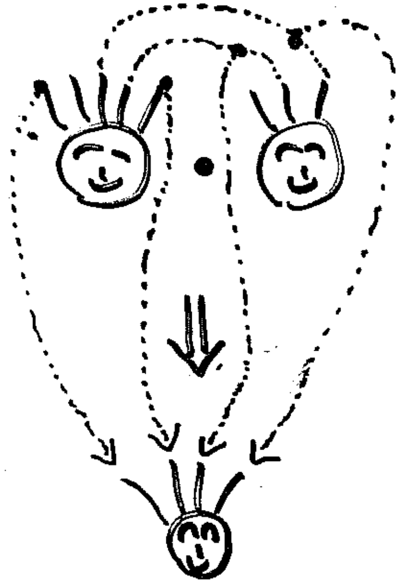
  $\lambda x.\, x+1$ ($NAT \to NAT$) • $True$ ($BOOL$) ×

- The underlying partial algebra of types controls program composability.

$$\begin{cases} (NAT \to NAT) \circledast NAT = NAT \\ (NAT \to NAT) \circledast BOOL = \text{Undefined} \end{cases}$$

Basic Idea (2)

- When it comes to processes, composition becomes:



cf.

$3 + 5$

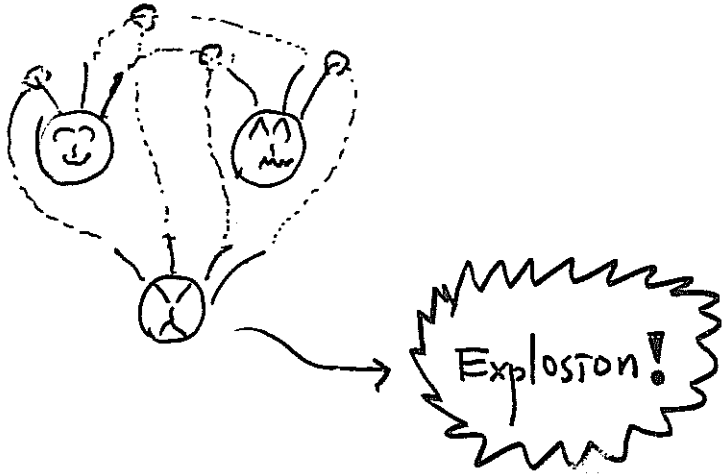
$\Downarrow$

8

$P \bullet_\delta \delta \xmapsto[\tau]{} \sigma$

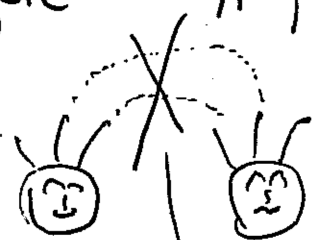
Basic Idea (3)

- But some composition is dangerous!



- divergence
- deadlock
- run-time error
- ⋮

- Therefore we type processes,



The connection is prohibited

Discussions

- What algebraic theories do we get from the name-free presentation?
- What semantic space does the present theory suggest for concurrent computation? (cf. Girard).
- Applications of symmetries:
  - Axiomatisation of "name passing".
  - Separation results by Palamidessi.
  - Game semantics.