A Domain Theory for Concurrency

Lectures by G. Winskel
(Cambridge)

- HOPLA and its denotational semantics

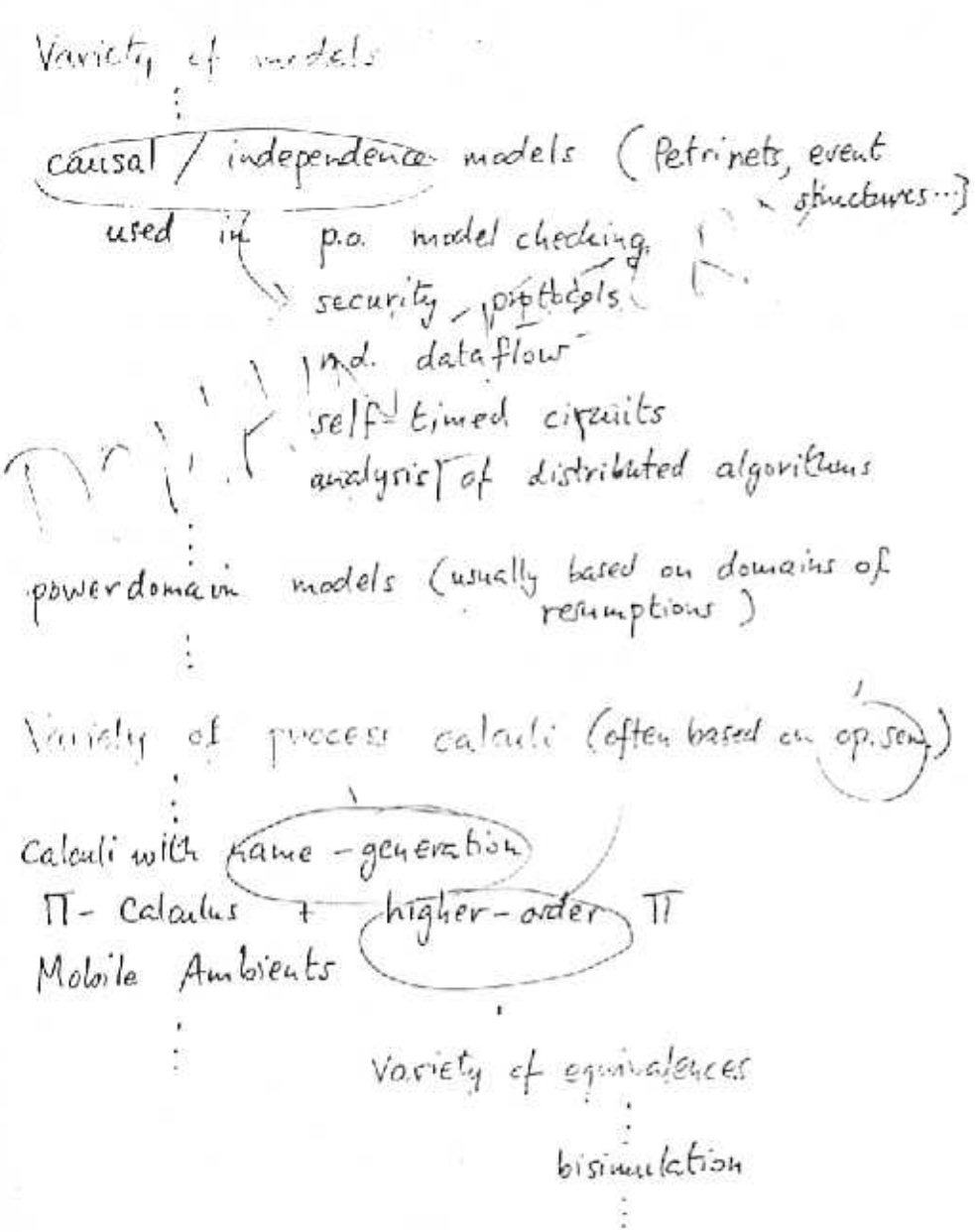
Glynn Winskel
Cambridge University Computer Laboratory.

Domain theory by providing a global mathematical setting for computation

- places PL's in connection with each other
- connects with algebra, topology & logic
- inspires PL's, type disciplines, reasoning methods

In concurrent/distributed computation that global mathematical guidance is missing ...

Theories of concurrency



Material www.cl.cam.ac.uk/~gw104

Primary:

"Domain theory for concurrency"
+ M. Nygaard

Supplementary:

"Profunctors, open maps & bisimulation"
+ G.L. Cattani

"Models for concurrency" + M. Nielsen

"Notes on category theory"
+ M. Caccamo (+ M. Hyland)

"Bisimulation from open maps"
+ M. Nielsen + A. Joyal

"Events in security protocols"
+ F. Crazzolara

Cts a category

- objects $\mathbb{P}, \mathbb{Q}, \ldots$ (pats orders)
- maps $\mathbb{P} \xrightarrow[cts]{} \mathbb{Q}$ are

continuous fns $\widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$

corr. to lin. maps $!\mathbb{P} \xrightarrow[Lin]{} \mathbb{Q}$

$!\mathbb{P}$ = p.o. of finite els. of $\widehat{\mathbb{P}}$

Adjunction: $Lin \underset{!}{\overset{\perp}{\rightleftarrows}} Cts$

$$Lin(!\mathbb{P}, \mathbb{Q}) \xrightarrow{\simeq} Cts(\mathbb{P}, \mathbb{Q})$$

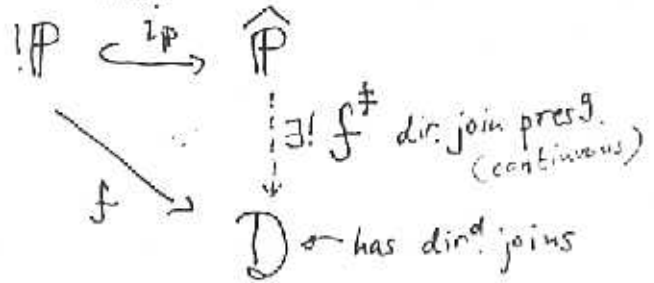
$$g \longmapsto g \circ \eta_{\mathbb{P}}$$

where $\eta_{\mathbb{P}} : \mathbb{P} \xrightarrow[cts]{} !\mathbb{P}$ s.t. $\eta_{\mathbb{P}}(X) = \{P \mid P \in !\mathbb{P} \& P \subseteq X\}$.

Cts has product $\mathbb{P} \& \mathbb{Q}$,

fn. space $(!\mathbb{P}) \multimap \mathbb{Q}$.

$\widehat{\mathbb{P}}$ is directed join completion of $!\mathbb{P}$ (ideal compl^n)



FACT: $\forall X \in \widehat{\mathbb{P}}.\quad X = \bigcup\{P \in !\mathbb{P} \mid P \subseteq X\}$ (directed join)

Uniqueness of $f^\sharp$:

$$f^\sharp(P) = f(P), \quad \text{all } P \in !\mathbb{P}$$

$$f^\sharp(X) = \bigcup\{f(P) \mid P \in !\mathbb{P} \ \& \ P \subseteq X\}$$

Existence of $f^\sharp$:

$$f^\sharp(\bigcup_{\text{directed}} X_i) = \bigcup\{f(P) \mid P \in !\mathbb{P} \ \& \ P \subseteq \bigcup X_i\}$$

$$= \bigcup_i \bigcup\{f(P) \mid P \in !\mathbb{P} \ \& \ P \subseteq X_i\} \quad \text{as join is directed}$$

$$= \bigcup_i f^\sharp(X_i)$$

Domains of non-det. processes

$\mathbb{P}$ pre-order of computation paths

e.g. L^* or L^+ ordered by extension $ab \leq abb$

A non-deterministic process (with paths) of type $\mathbb{P}$ is a monotonic fn. $X : \mathbb{P}^{op} \to 2$ (1 ↦ true, 0 ↦ false)

Non-det. processes of type $\mathbb{P}$ form domain

$$\widehat{\mathbb{P}} = [\mathbb{P}^{op}, 2]$$ (path sets)

Iso. to $\leq$-downclosed subsets of $\mathbb{P}$ under $\subseteq$.

Has nd. sums given by $\cup$.

The free join completion of $\mathbb{P}$.

E.g. if $\mathbb{P} = L^+$, $\widehat{\mathbb{P}}$ = Hoare powerdomain.

We'll identify $\widehat{\mathbb{P}}$ with path sets ordered by $\subseteq$.

Lin a linear category:

- objects po's of paths $\mathbb{P}, \mathbb{Q}, \ldots$
- maps $\mathbb{P} \xrightarrow{}_{Lin} \mathbb{Q}$ are join preserving functions $\widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$

corr. to monotonic functions $\mathbb{P} \to \widehat{\mathbb{Q}}$

corr. to monotonic functions $\mathbb{P} \times \mathbb{Q}^{op} \to 2$ (relations) $\widehat{\mathbb{P}^{op} \times \mathbb{Q}}$

Has tensor $\mathbb{P} \times \mathbb{Q}$, fn space $\mathbb{P}^{op} \times \mathbb{Q}$,

product & coproduct $\mathbb{P} + \mathbb{Q}$

A useful alternative defn. of the exponential $!\mathbb{P}$

$!\mathbb{P}$ = preorder of $\mathcal{Y}_{fin}(\mathbb{P})$ ordered by

$$P \preceq P' \iff \forall p \in P\ \exists p' \in P'.\ p \leq_{\mathbb{P}} p'.$$

NB. $\widehat{!\mathbb{P}} \cong$ 

Exercise (Characterising the domains)

A complete lattice $\mathcal{D}$ is prime algebraic when for all $d \in \mathcal{D}$

$$d = \bigsqcup \{ p \sqsubseteq d \mid p \text{ is a complete prime of } \mathcal{D} \}$$

An element $p \in \mathcal{D}$ is a complete prime iff for all $X \subseteq \mathcal{D}$

$$p \sqsubseteq \bigsqcup X \Rightarrow \exists x \in X.\ p \sqsubseteq x.$$

(i) For a partial order P show $\widehat{P}$ is a prime algebraic complete lattice.

(ii) Show any prime algebraic complete lattice is isomorphic to $\widehat{P}$ where P is the order of $\mathcal{D}$ restricted to the complete primes.

Type	Term	Denotation
all types	$\sum_{i\in I} t_i$, $\mathrm{rec}\, x\, t$	union, fixed pt.
$!\mathbb{P}$	prefix $!t$	denotes $\eta_{\mathbb{P}}(t)$ where $\eta_{\mathbb{P}} : \mathbb{P} \to_{cb} !\mathbb{P}$
	match $[u > !x \Rightarrow t]$	denotes $\bar{t}(u)$ where $!\mathbb{P} \xrightarrow{\bar{t}} \mathbb{Q}$ transpose of $\mathbb{P} \xrightarrow{t} \mathbb{Q}$
$\mathbb{P} \to \mathbb{Q}$	$\lambda x\, t$, $t\, v$	abstractn., applicn.
$\sum_{\alpha\in A} \alpha!\mathbb{P}_\alpha$	αt, $\pi_\alpha(u)$	injection, projn.
$\mu_j \vec{\mathbb{P}}\, \vec{\mathbb{T}}$	$\mathrm{abs}(t)$, $\mathrm{rep}(u)$	iso $\mathbb{T}_j[\mu \vec{\mathbb{P}}\, \vec{\mathbb{T}}/\vec{\mathbb{P}}] \cong \mu_j \vec{\mathbb{P}}\, \vec{\mathbb{T}}$ (abs, rep)

Example: Type for CCS + process passing

$$\mu P.\ \tau!P + \sum_{a\in A} a!(P \to P) + \sum_{a\in\bar{A}} \bar{a}!(1P + 2P)$$

$$\mathbb{T} ::= \mathbb{T}_1 \to \mathbb{T}_2 \quad \Big|\ \sum_{\alpha\in A} \mathbb{T}_\alpha \quad \Big|\ !\mathbb{T} \quad \Big|\ \mathbb{T} \quad \Big|\ \mu_j \vec{\mathbb{T}}.\ \vec{\mathbb{T}}$$

Closed type expressions $\mathbb{P}$ denote path orders with paths $p : \mathbb{P}$ ordered $p \le_{\mathbb{P}} p'$:

$$\frac{p : !\mathbb{P} \qquad q : \mathbb{Q}}{p \mapsto q : \mathbb{P} \to \mathbb{Q}} \qquad \frac{p' \le_{!\mathbb{P}} p \qquad q \le_{\mathbb{Q}} q'}{p \mapsto q \ \le_{\mathbb{P}\to\mathbb{Q}}\ p' \mapsto q'}$$

$$\frac{p : \mathbb{P}_\beta \qquad \beta \in A}{\beta p : \sum_{\alpha\in A} \mathbb{P}_\alpha} \qquad \frac{p \le_{\mathbb{P}_\beta} p'}{\beta p \ \le_{\sum_\alpha \mathbb{P}_\alpha}\ \beta p'}$$

$$\frac{p_1 : \mathbb{P}, \ \cdots, \ p_n : \mathbb{P}}{\{p_1, \cdots, p_n\} : !\mathbb{P}} \qquad \frac{p \le_{\mathbb{P}} p'}{p \le_{!\mathbb{P}} p'}$$

$$\frac{p : \mathbb{T}_j[\mu \vec{T}.\ \vec{\mathbb{T}}/\vec{T}]}{\mathrm{abs}\, p : \mu_j \vec{T}.\ \vec{\mathbb{T}}} \qquad \frac{p \le_{\mathbb{T}_j[\mu\cdots]} p'}{\mathrm{abs}\, p \le_{\mu_j \vec{T}.\vec{\mathbb{T}}} \mathrm{abs}\, p'}$$

$$\frac{\Gamma, x:\mathbb{P} \vdash t:\mathbb{P}}{\Gamma \vdash \nu x.t : \mathbb{P}} \qquad \frac{\Gamma \vdash t_j : \mathbb{P} \text{ all } j \in I}{\Gamma \vdash \sum_{i\in I} t_i : \mathbb{P}}$$

$$\frac{\Gamma, x:\mathbb{P} \vdash t:\mathbb{Q}}{\Gamma \vdash \lambda x.t : \mathbb{P}\to\mathbb{Q}} \qquad \frac{\Gamma \vdash t:\mathbb{P}\to\mathbb{Q} \quad \Lambda \vdash u:\mathbb{P}}{\Gamma, \Lambda \vdash tu : \mathbb{Q}}$$

$$\frac{\Gamma \vdash t:\mathbb{P}_\beta \quad \beta\in A}{\Gamma \vdash \beta t : \sum_{\alpha\in A} \mathbb{P}_\alpha} \qquad \frac{\Gamma \vdash t : \sum_{\alpha} \mathbb{P}_\alpha \quad \beta\in A}{\Gamma \vdash \pi_\beta t : \mathbb{P}_\beta}$$

$$\frac{\Gamma \vdash u:\mathbb{P}}{\Gamma \vdash !u : !\mathbb{P}} \qquad \frac{\Gamma, x:\mathbb{P} \vdash t:\mathbb{Q} \quad \Lambda \vdash u : !\mathbb{P}}{\Gamma, \Lambda \vdash [u > !x \Rightarrow t] : \mathbb{Q}}$$

$$\frac{\Gamma \vdash t : T_j[\mu \vec{T}.\vec{T}/\vec{T}]}{\Gamma \vdash \text{abs}\, t : \mu_j \vec{T}.\vec{T}} \qquad \frac{\Gamma \vdash t : \mu_j \vec{T}.\vec{T}}{\Gamma \vdash \text{rep}\, t : T_j[\mu\cdot\cdot/\vec{T}]}$$

+ structural rules

[Mikkel Nygaard, gur]

Higher Order Process LAnguage

Types	Terms	Actions
all	$\sum_{i\in I} t_i$ $\quad \mathrm{rec}\, x\, t$	
$!\mathbb{P}$	$!t$ $[u > !x \Rightarrow t]$	"beep!"
$\mathbb{P}\to\mathbb{Q}$	$\lambda x\, t$ $t\, u$	$u \mapsto a$
$\sum_{\alpha\in A} \mathbb{P}_\alpha$	αt $\pi_\alpha(u)$	$\alpha\, a$
$\mu_j \vec{P}\,\vec{\mathbb{P}}$		$\mathrm{abs}\, a$

Eg. Process-Passing CCS

$\mu P \quad \tau.!P + \sum_{\alpha\in A} \alpha!(P\to P) + \sum_{\alpha\in A} \bar{\alpha}!(1P + 2P)$

Mobile Ambients w. Public Names ...

Actions

$$a ::= ! \mid u \mapsto a \mid \alpha\, a$$

$$\frac{}{!\mathbb{P} : ! : \mathbb{P}}$$

$$\frac{u : \mathbb{P} \qquad \mathbb{Q} : a : \mathbb{Q}'}{\mathbb{P}\to\mathbb{Q} : u\mapsto a : \mathbb{Q}'}$$

$$\frac{\mathbb{P}_\beta : a : \mathbb{P}'}{\sum_{\alpha\in A} \alpha\, \mathbb{P}_\alpha : \beta\, a : \mathbb{P}'}$$

$$\frac{\mathbb{P}_j[\mu \vec{P}\,\vec{\mathbb{P}}/\vec{P}] : a : \mathbb{P}'}{\mu_j \vec{P}\,\vec{\mathbb{P}} : \mathrm{abs}\, a : \mathbb{P}'}$$

$\mathbb{P} : t \xrightarrow{a} t'$ & $\mathbb{P} : a : \mathbb{P}' \Rightarrow t' : \mathbb{P}'$

$\rightsquigarrow$

$\mathbb{P} : t \xrightarrow{a} t' : \mathbb{P}'$

$$\frac{\mathbb{P}: t[\mathrm{rec}\, x\, t/x] \xrightarrow{a} t'}{\mathbb{P}: \mathrm{rec}\, x\, t \xrightarrow{a} t'} \qquad \frac{\mathbb{P}: t_j \xrightarrow{a} t'}{\mathbb{P}: \sum_{i\in I} t_i \xrightarrow{a} t'}$$

$$!\mathbb{P}: !t \xrightarrow{!} t \qquad \frac{!\mathbb{P}: u \xrightarrow{!} u' \quad \mathbb{Q}: t[u'/x] \xrightarrow{a} t'}{\mathbb{Q}: [u > !x \Rightarrow t] \xrightarrow{a} t'}$$

$$\frac{\mathbb{Q}: t[u/x] \xrightarrow{a} t'}{\mathbb{P}\to\mathbb{Q}: \lambda x\, t \xrightarrow{u\mapsto a} t'} \qquad \frac{\mathbb{P}\to\mathbb{Q}: t \xrightarrow{u\mapsto a} t'}{\mathbb{Q}: tu \xrightarrow{a} t'}$$

$$\frac{\mathbb{P}_\beta: t \xrightarrow{a} t'}{\sum_{\alpha\in A} \mathbb{P}_\alpha: \beta t \xrightarrow{\beta a} t'} \qquad \frac{\sum_{\alpha\in A} \mathbb{P}_\alpha: t \xrightarrow{\beta a} t'}{\mathbb{P}_\beta: \pi_\beta(t) \xrightarrow{a} t'}$$

$$\frac{[\mu\vec{P}\vec{\mathbb{P}}/\vec{P}]: t \xrightarrow{a} t'}{\mu_j\vec{P}\vec{\mathbb{P}}: \mathrm{abs}(t) \xrightarrow{\mathrm{abs}(a)} t'} \qquad \frac{\mu_j\vec{P}\vec{\mathbb{P}}: t \xrightarrow{\mathrm{abs}(a)} t'}{\mathbb{P}_j[\mu\vec{P}\vec{\mathbb{P}}/\vec{P}]: \mathrm{rep}(t) \xrightarrow{a} t'}$$

Products

$\mathbb{P} \,\&\, \mathbb{Q} \equiv_{def} 1\,\mathbb{P} + 2\,\mathbb{Q}$ product type

$(t,u) \equiv_{def} 1\,t + 2\,u$ pairs

$fst(t) \equiv_{def} \pi_1(t)$ $\quad snd(t) \equiv_{def} \pi_2(t)$ projections

$(p,-) \equiv_{def} 1p$ $\quad (-,q) \equiv_{def} 2q$ actions

Derive:

$$\frac{\Gamma \vdash t:\mathbb{P} \quad \Gamma \vdash u:\mathbb{Q}}{\Gamma \vdash (t,u):\mathbb{P}\&\mathbb{Q}} \qquad \frac{\Gamma \vdash t:\mathbb{P}\&\mathbb{Q}}{\Gamma \vdash fst(t):\mathbb{P}}$$

$$\frac{P: t \xrightarrow{p} t'}{\mathbb{P}\&\mathbb{Q}: (t,u) \xrightarrow{(p,-)} t'} \qquad \frac{\mathbb{Q}: u \xrightarrow{q} u'}{\mathbb{P}\&\mathbb{Q}: (t,u) \xrightarrow{(-,q)} u'}$$

$$\frac{\mathbb{P}\&\mathbb{Q}: t \xrightarrow{(p,-)} t'}{\mathbb{P}: fst(t) \xrightarrow{p} t'} \qquad \frac{\mathbb{P}\&\mathbb{Q}: t \xrightarrow{(-,q)} t'}{\mathbb{Q}: snd(t) \xrightarrow{q} t'}$$

Prefixed sums

$$\sum_{\alpha \in A} \alpha . P_\alpha \equiv_{def} \sum_{\alpha \in A} !P_\alpha$$

$$\beta . t \equiv_{def} \beta (!t)$$

$$[u > \beta . x \Rightarrow t] \equiv_{def} [\pi_\beta(u) > !x . \Rightarrow t]$$

Types & transitions:

A type respecting relation $\mathcal{R}$ on closed process terms is a collection

$\mathcal{R}_{\mathbb{P}}$ indexed by types $\mathbb{P}$

such that

if $t_1 \mathcal{R}_{\mathbb{P}} t_2$ then $t_1 : \mathbb{P}$ and $t_2 : \mathbb{P}$.

A bisimulation is a type respecting relⁿ $\mathcal{R}$ s.t. whenever $t_1 \mathcal{R}_{\mathbb{P}} t_2$,

$\mathbb{P} : t_1 \xrightarrow{a} t_1' : \mathbb{Q} \Rightarrow \exists t_2' .\ \mathbb{P} : t_2 \xrightarrow{a} t_2' : \mathbb{Q}$ &

$t_1' \mathcal{R}_{\mathbb{Q}} t_2'$,

$\mathbb{P} : t_2 \xrightarrow{a} t_2' : \mathbb{Q} \Rightarrow \exists t_1' .\ \mathbb{P} : t_1 \xrightarrow{a} t_1' : \mathbb{Q}$ &

$t_1' \mathcal{R}_{\mathbb{Q}} t_2'$.

$\sim$ the largest bisimulation

Useful bisimilarities :

$$rec\ x\ t \sim t[rec\ x\ t / x]$$

$$[!u > !x \Rightarrow t] \sim t[u/x]$$

$$[\sum_{i\in I} u_i > !x \Rightarrow t] \sim \sum_{i\in I} [u_i >. x \Rightarrow t]$$

$$[u > !x \Rightarrow \sum_{i\in I} t_i] \sim \sum_{i\in I} [u > !x \Rightarrow t_i]$$

$$(\lambda x\, t)\ u \sim t[u/x]$$

$$\lambda x\ (t\ x) \sim t$$

$$\lambda x\ (\sum_{i\in I} t_i) \sim \sum_{i\in I} (\lambda x\ t_i)$$

$$(\sum_i t_i)\ u \sim \sum_i (t_i\ u)$$

$$\pi_a(a.t) \sim t$$

$$\pi_a(b.t) \sim nil \qquad b \neq a$$

$$t \sim \sum_{a\in A} a.\pi_a(t)$$

$$\pi_a(\sum_i t_i) \sim \sum_i \pi_a(t_i)$$

Derived bisimilarities :

$$[a.u > a.x \Rightarrow t] \sim t[u/x]$$

$$[b.u > a.x \Rightarrow t] \sim nil \qquad a \neq b$$

$$[\sum_i u_i > a.x \Rightarrow t] \sim \sum_i [u_i > a.x \Rightarrow t]$$

$$[u > a.x \Rightarrow \sum_i t_i] \sim \sum_i [u > a.x \Rightarrow t_i]$$

$$fst(t,u) \sim t$$

$$snd(t,u) \sim u$$

$$t \sim (fst(t), snd(t))$$

$$(\sum_{i\in I} t_i, \sum_{i\in I} u_i) \sim \sum_{i\in I} (t_i, u_i)$$

$$fst(\sum_{i\in I} t_i) \sim \sum_{i\in I} fst(t_i)$$

$$snd(\sum_{i\in I} t_i) \sim \sum_{i\in I} snd(t_i)$$

Theorem:

$\sim$ is a congruence

Proof: A method due to Doug Howe.

NON-EXAMINABLE But fairly detailed sketch in the notes. ⊠

Proposition: Let $t_1, t_2 : \mathbb{P} \to \mathbb{Q}$.
T.f.a.e.:

1. $t_1 \sim t_2$
2. $t_1 u \sim t_2 u$ for all $u : \mathbb{P}$
3. $t_1 u_1 \sim t_2 u_2$ for all $u_1 \sim_{\mathbb{P}} u_2$

Proof $1 \Rightarrow 2 \& 3$ by congruence.
$3 \Rightarrow 2$ obvious
$2 \Rightarrow 1$: use $t \xrightarrow{u:\mathbb{P}} t'$ iff $tu \xrightarrow{a} t'$ ⊠

Linearity

$x : \mathbb{P} \vdash t : \mathbb{Q}$
determines a function (operation)
$u : \mathbb{P} \longmapsto t[u/x] : \mathbb{Q}$.

Informally, the function is linear when any output transition of $t[u/x]$ requires precisely one execution of u.

Formally, the term t is linear in x when

$$t\left[\sum_{i\in I} u_i / x\right] \sim \sum_{i\in I} t[u_i/x]$$

for all sums $\sum_{i\in I} u_i : \mathbb{P}$.

Non-linear terms:

- $x : \mathbb{P} \vdash !x : !\mathbb{P}$
- $!x > a.y \Rightarrow [x > b.z \Rightarrow c.nil.!]$

Recursive declarations in HML, eg. CCS

$$P \overset{def}{=} \tau.P + \sum_{a \in L} \bar{a}.P + \sum_{a \in L} a.P$$

by

$$P \equiv \mu P\,(\tau.P + \sum_{a} \bar{a}.P + \sum_{a} a.P)$$

$$X \| Y \overset{def}{=}$$

$$\sum_{\alpha \in L \cup \bar{L} \cup \{\tau\}} [X > \alpha.x \Rightarrow \alpha.(x \| Y)] +$$

$$\sum_{\alpha} [Y > \alpha.y \Rightarrow \alpha.(X \| x)] +$$

$$\sum_{a \in L \cup \bar{L}} [X > a.x \Rightarrow [Y > \bar{a}.y \Rightarrow \tau.(x \| y)]]$$

by

$$\| \equiv rec\ P\ \lambda x\ \lambda y$$

$$\sum_{\alpha} [X > \alpha.x \Rightarrow \alpha.\ P x Y] +$$

$$\sum_{\alpha} [Y > \alpha.y \Rightarrow \alpha.\ P X y] +$$

$$\sum_{a} [X > a.x \Rightarrow [Y > \bar{a}.y \Rightarrow \tau.\ P x y]]$$

Deriving the expansion theorem of CCS:

$$P \sim \sum_{\alpha} \sum_{i \in I(\alpha)} \alpha.P_i \qquad Q \sim \sum_{\alpha} \sum_{j \in J(\alpha)} \alpha.Q_j$$

$$P \| Q \sim$$

$$\sum_{\alpha} [P > \alpha.x \Rightarrow \alpha.(x \| Q)] +$$

$$\sum_{\alpha} [Q > \alpha.y \Rightarrow \alpha.(P \| y)] +$$

$$\sum_{a} [P > a.x \Rightarrow [Q > \bar{a}.y \Rightarrow \tau.(x \| y)]]$$

$$\sim$$

$$\sum_{\alpha} \sum_{i \in I(\alpha)} \alpha.(P_i \| Q) +$$

$$\sum_{\alpha} \sum_{i \in J(\alpha)} \alpha.(P \| Q_j) +$$

$$\sum_{a} \sum_{i \in I(a)} \sum_{j \in J(a)} \tau.(P_i \| Q_j)$$

by props. of ~~+~~ $\sum$ match.

Higher-Order CCS — process passing.

$a!(t,u)$ $\quad a?(\lambda x\ t)$

$$\mathbb{P} = \tau.\mathbb{P} + \sum_a a!.\mathbb{C} + \sum_a a?.\mathbb{F}$$

$$\mathbb{C} = \mathbb{P} \,\&\, \mathbb{P}$$

$$\mathbb{F} = \mathbb{P} \to \mathbb{P}$$

$$P \| Q = \sum_\alpha [P > \alpha.x \Rightarrow \alpha.(x \| Q)] + \sum_\alpha [Q > \alpha.y \Rightarrow \alpha.(P \| y)] +$$

$$\sum_a [P > a?.f \Rightarrow [Q > a!.c \Rightarrow \tau.(f \| c)]] +$$

$$\sum_a [P > a!.c \Rightarrow [Q > a?.f \Rightarrow \tau.(c \| f)]]$$

$$F \| C = (F\ \mathrm{fst}\ C) \| \mathrm{snd}\ C$$

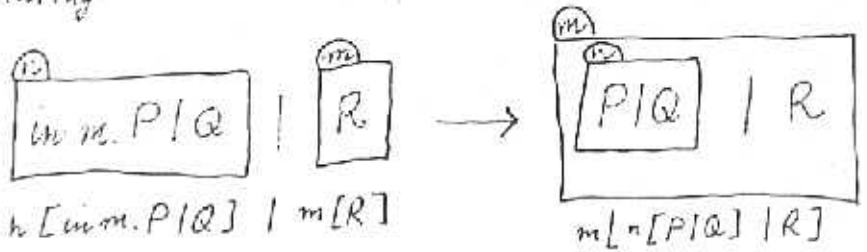
$$F \| P = \lambda x\ (Fx \| P)$$

$$C \| P = (\mathrm{fst}\ C, (\mathrm{snd}\ C) \| P)$$

Mobile Ambients

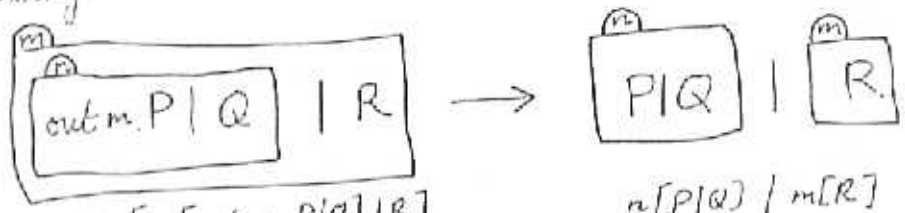
72

Entering:



$n[\mathrm{in}\ m.P|Q] \mid m[R]$ $\qquad$ $m[n[P|Q] \mid R]$

Exiting:



$m[n[\mathrm{out}\ m.P|Q] \mid R]$ $\qquad$ $n[P|Q] \mid m[R]$

Opening:

$\mathrm{open}\ m!\ P \mid m[Q]$ $\qquad$ $P|Q$

$$m[P] =$$

$$\text{open } m?.P \quad +$$

$$\text{mvin } m?.(\lambda y \; m[y \mid P]) \quad +$$

$$[P > \tau.x \Rightarrow \tau.\, m[x]] \; +$$

$$\sum_n [P > \text{in}_n.x \Rightarrow \text{mvin } n!.(m[x], nil.)] +$$

$$\sum_n [P > \text{out}_n.x \Rightarrow \text{mvout } n!.(m[x], nil) +$$

$$[P > \text{mvout } m!.z \Rightarrow \tau.((\text{fst } z) \mid m[\text{snd } z])]$$

So ambient creation $m[P]$ is derived as a recursively defined operator in HOPLA.

	operational	denotational
(1) Transitions	$\mathbb{P} : t \xrightarrow{a}$.	$[\![a^*(t)]\!] \neq \emptyset$
(2) Transitions	$!\mathbb{P} : t \xrightarrow{!}$.	$[\![t]\!] \neq \emptyset$
(3) $!\mathbb{O}$-Transitions (typo)	$!\mathbb{C} : t \xrightarrow{!}$.	$[\![t]\!] \neq \emptyset$

Can reduce to $!\mathbb{O}$-transitions within a suitable context:

(1) to (2): $\mathbb{P}: t \xrightarrow{a} t' : \mathbb{P}'$ iff $!\mathbb{P}' : a^*(t) \xrightarrow{!}$

where $(u \mapsto a)^*(t) \equiv a^*(tu)$ $\quad (\beta a)^*(t) \equiv a^*(\pi_\beta t)$

$!^*(t) \equiv t$ $\quad (\mathrm{abs}\ a)^*(t) \equiv a^*(\mathrm{rep}\ t)$.

(2) to (3):

$!\mathbb{P} : t \xrightarrow{!}$ iff $!\mathbb{O} : C_0[t] \xrightarrow{!}$

where $C_0[-] \equiv [- > !x \Rightarrow !\emptyset]$.

For $t : {}^{\circ}!\mathbb{P}$, $[\![t]\!] \neq \emptyset \Leftrightarrow t \xrightarrow{!}$.

Or equivalently:

For $t : \mathbb{P}$, $[\![a^*(t)]\!] \neq \emptyset \Leftrightarrow t \xrightarrow{a}$.

"$\Leftarrow$" By soundness, ie.

$\mathbb{P} : t \xrightarrow{a} t' : \mathbb{P}' \Rightarrow [\![a^*(t)]\!] \supseteq \eta_{\mathbb{P}'}[\![t']\!]$

Proof. By ind. on derivations.

"$\Rightarrow$" By 'logical relations,'

For $t : \mathbb{P}$, $[\![t]\!] \trianglelefteq_{\mathbb{P}} t$, where

---

$X \trianglelefteq_{\mathbb{P}} t \Leftrightarrow \forall p \in X.\ p\ \varepsilon_{\mathbb{P}}\ t$

$p \mapsto q\ \varepsilon_{\mathbb{P}\to\mathbb{Q}}\ t \Leftrightarrow \forall u.\ (P \trianglelefteq_{\mathbb{P}} u \Rightarrow q\ \varepsilon_{\mathbb{Q}}\ tu)$

$\beta p\ \varepsilon_{\sum_{\alpha} \mathbb{P}_\alpha}\ t \Leftrightarrow p\ \varepsilon_{\mathbb{P}_\beta}\ \pi_\beta t$

$p\ \varepsilon_{!\mathbb{P}}\ t \Leftrightarrow \exists t'.\ !\mathbb{P} : t \xrightarrow{!} t' \ \&\ P \trianglelefteq_{\mathbb{P}} t'$

$\mathrm{abs}\, p\ \varepsilon_{\mu_j \vec{T}.\vec{T}}\ t \Leftrightarrow p\ \varepsilon_{T_j[\mu \vec{T}.\vec{T}/\vec{T}]}\ \mathrm{rep}\ t$.

$P \in !\mathbb{P}$ consists of several, possibly no, computation paths [a compound computation path associated with running several copies of a process].

While discarding is easy, copying is often restricted in a distributed computation.

↳ Many operations of dist. comp have the following property ('Affine linearity'):

A computation path of the process arising from the operation applied to an input process has resulted from at most one computation path of the input process.

Full Abstraction

$[\![t_1]\!] \subseteq [\![t_2]\!]$ iff

for all contexts $C[-] : !\mathbb{O}$

$C[t_1] \xrightarrow{!} \Rightarrow C[t_2] \xrightarrow{!}$.

'only if' compositionality of den. sem. + adequacy.

'if' For all $p : \mathbb{P}$ there is $C_p[-] : !\mathbb{O}$

$\forall t : \mathbb{P}$ $p \in [\![t]\!] \Leftrightarrow [\![C_p[t]]\!] \neq \emptyset$

i.e. $C_p[t] \xrightarrow{!}$

Eg. $C_\phi[-] \equiv [- > !x \Rightarrow !\phi]$.

Aff an affine linear category

objects $\mathbb{P}, \mathbb{Q}, \cdots$ p.o's

maps $\mathbb{P} \xrightarrow[Aff]{} \mathbb{Q}$ are

non-empty-join preserving fns. $\widehat{\mathbb{P}} \longrightarrow \widehat{\mathbb{Q}}$

corr. to linear maps $\mathbb{P}_\perp \longrightarrow \mathbb{Q}$

An affine HOPLA with types & terms:

$\mathbb{P}_\perp = \overset{\mathbb{P}}{\overset{\vee}{\underset{\perp}{\bullet}}}$ $\quad t \;,\; [u > x \Rightarrow t]$

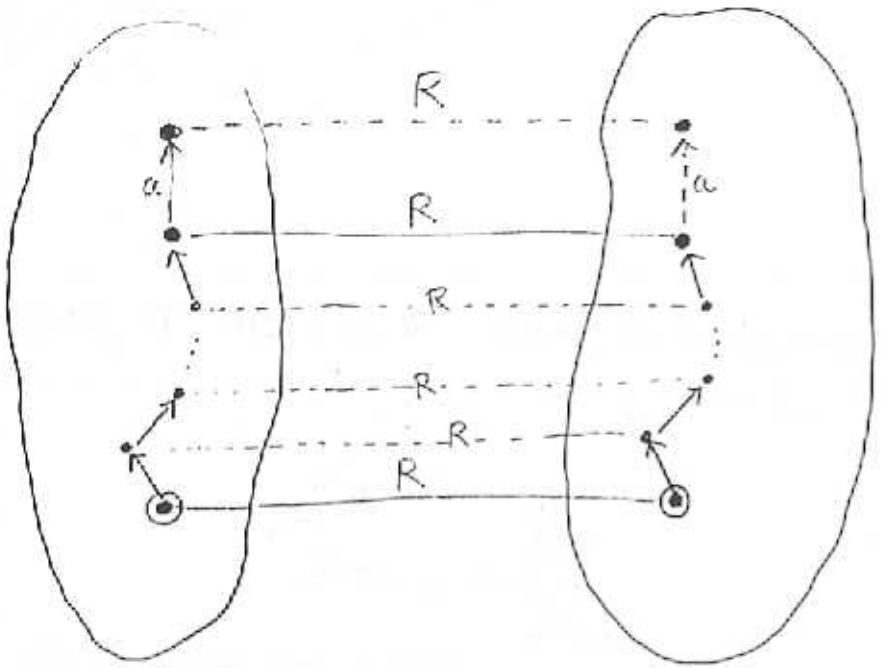
$\mathbb{P} \otimes \mathbb{Q} = \mathbb{P}_\perp \times \mathbb{Q}_\perp \setminus \{(\perp,\perp)\}$ $\quad t \otimes u \;,\; [u > x \otimes y \Rightarrow t]$

$\mathbb{P} \multimap \mathbb{Q} = (\mathbb{P}_\perp)^{op} \times \mathbb{Q}$ $\quad \lambda x\, t \;,\; tu$

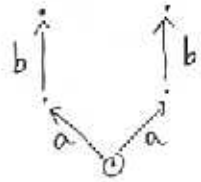
$\sum_{a \in A} \mathbb{P}_a$ $\quad a t \;,\; \pi_a(u)$

recursive types. sums, recursion. [Nygaard+gw lics 04]

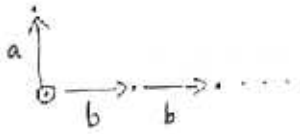
(Strong) bisimulation on transition systems



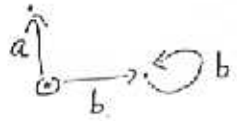
A bisimulation corresponds to a relation between comput.ⁿ paths which matches extensions of paths in one t.s. by extensions in the other.



bisimilar to

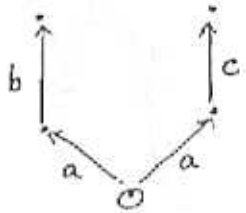


bisim. to

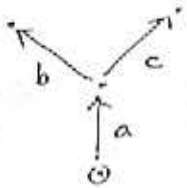


"

not bisim. to



not bisim. to



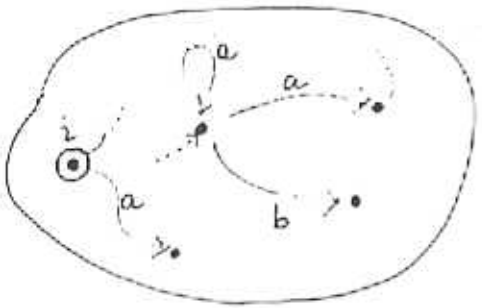
A transition system consists of

$$T = (S, i, L, \{\xrightarrow{a}\}_{a \in L})$$

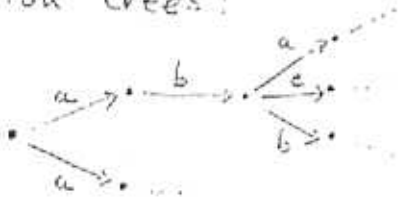
where S is a set of states, with state i

L is a set of labels

$s \xrightarrow{a} s'$ is the transition relation on states



Synchronisation trees:



... $\sigma : T \longrightarrow T'$ are

$\sigma : S \to S'$ s.t. $\sigma(i) = i'$ & $s \xrightarrow{a} s' \Rightarrow \sigma(s) \xrightarrow{a} \sigma(s')$.

Morphisms of transition systems [in a fibre]:

A morphism $f : T \longrightarrow T'$

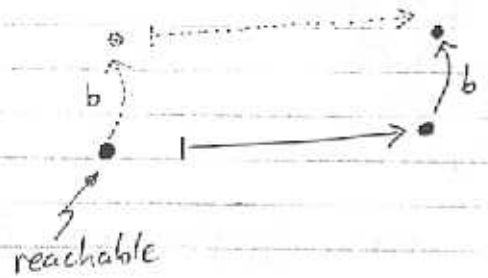
is a function $f : S \to S'$ s.t.

$$f(i) = i' \quad \&$$

$$s \xrightarrow{a} s' \Rightarrow f(s) \xrightarrow{a} f(s').$$

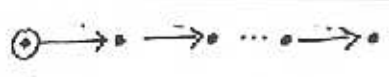
$f : T \to T'$ is a functional bisimulation

iff

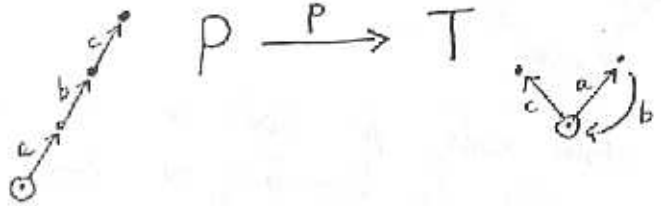


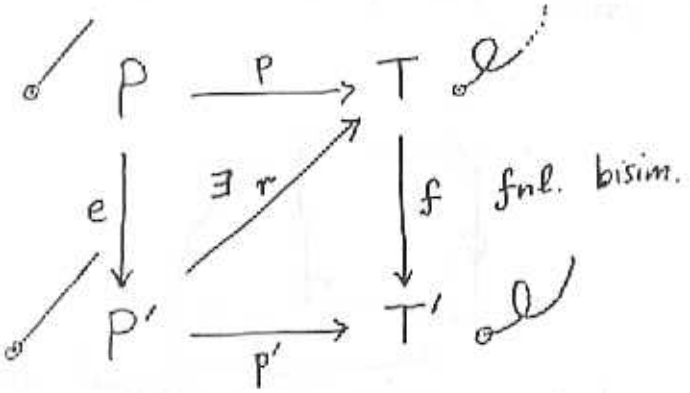
T_1, T_2 bisimilar

iff there is a "zig-zag" of fnl. bisims.

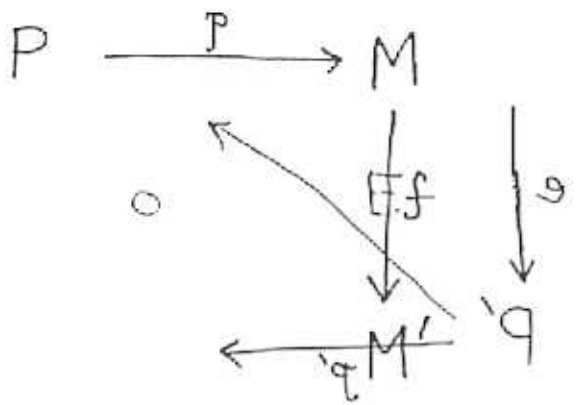
Paths "strings 

$\mathbb{P} \hookrightarrow \mathbb{T}$



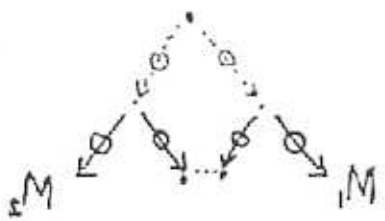


A general defn. of bisimulation



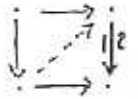
f is open iff ...

M_1, M_2 are $\mathbb{P}$-bisimilar iff

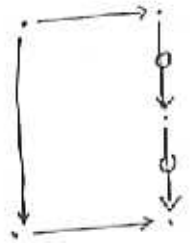


Basic properties of open maps:

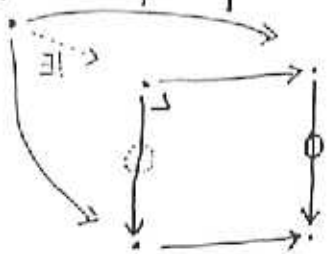
- isomorphisms are open



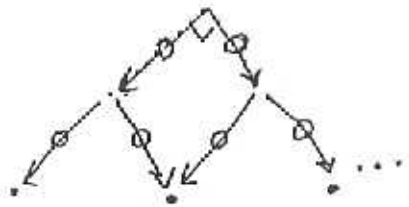
- a composition of opens is open

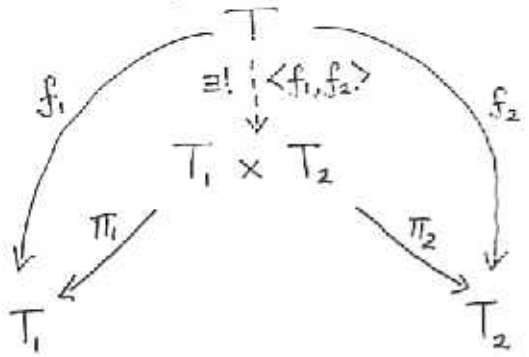


- a pullback of open is open



$\Rightarrow$ a "zig-zag" of open maps yields a span of open maps





$T_1 \times T_2 = (S_1 \times S_2, (i_1, i_2), tran)$

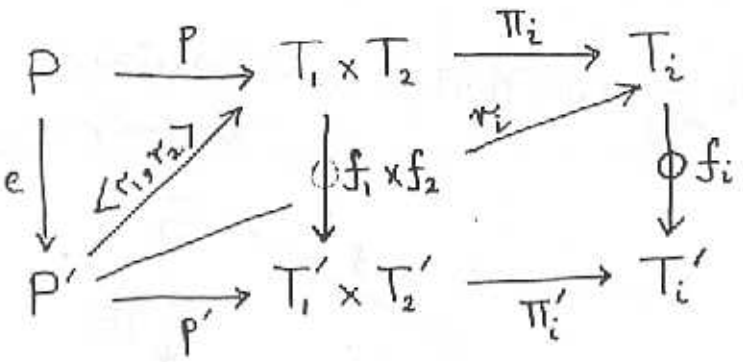
$(s_1, s_2) \xrightarrow{a} (s_1', s_2')$ in $T_1 \times T_2$

iff $s_1 \xrightarrow{a} s_1'$ in T_1 and

$s_2 \xrightarrow{a} s_2'$ in T_2.

π_1, π_2 project to 1st, 2nd coord.

The product of open maps is open [So product respects bisimulation].



Categories

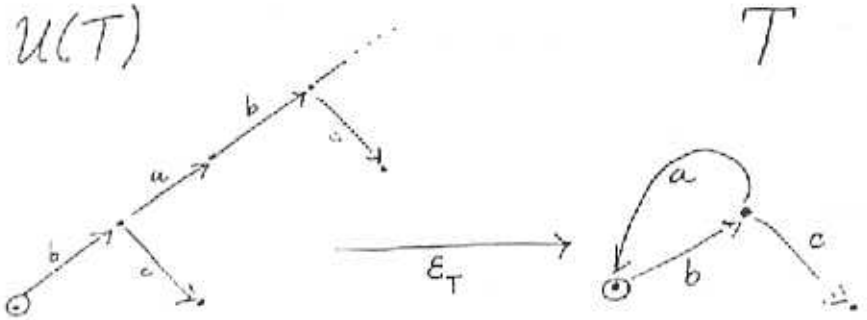
$\mathbb{T}$ of transition systems

$\mathbb{S}$ of synchronisation trees.

Relating models, an example

$\mathbb{S} \underset{T}{\overset{\mathcal{U}}{\rightleftarrows}} \mathbb{T}$ a coreflection

$\mathcal{U}$ unfolds a trans. system to its associated synchronisation tree:

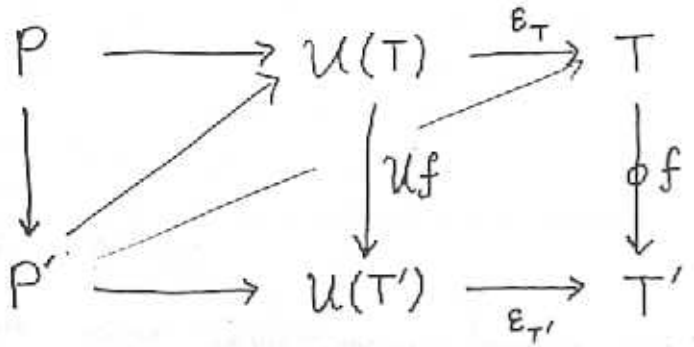


$$\begin{array}{ccc} \mathcal{U}(T) & \xrightarrow{\varepsilon_T} & T \\ \exists! \uparrow & \nearrow_{f} & \\ S & & \end{array}$$

$\Rightarrow$ adjunction

coreflection $\because$ $\mathcal{U}\mathcal{U}(T) \overset{\varepsilon_{\mathcal{U}T}}{\cong} \mathcal{U}(T)$

$\mathcal{U}$ preserves open maps (so bisim.)



Interleaving models — Independence models

Trans. sys. — Petri nets / Transition systems with independence

coreflections

Sign. trees — Event structures / Mazurkiewicz trace lang. / Occurrence nets

- Operations understood uniformly as universal constructions
- Preservation properties of adjoints help relate semantics
- Categories of models support an abstract / general definition of bisimulation.

Computations:

Sequences/strings/branches — Pomsets

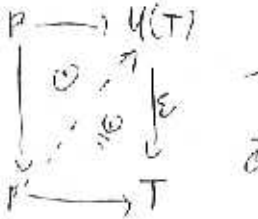
But,
what about models & bisimulation in general, eg. for value-passing or higher-order processes?

Need a generous space of models.

One answer: Presheaf models

$$\mathbb{P} \xrightarrow{\text{Yoneda}} [\mathbb{P}^{op}, Set]$$

an individual presheaf $X : \mathbb{P}^{op} \to Set$ can be thought of as a transition system with computation paths having shape in $\mathbb{P}$.



Exercise. Show

$$U(T) \xrightarrow{a_T} T$$

is open.