INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

Workshop QSLC
September the 2nd

université PARIS DIDEROT PARIS 7

Probabilistic Call by Push Value

joint work with **Thomas Ehrhard**

Christine Tasson
Christine.Tasson@pps.univ-paris-diderot.fr

Laboratoire IRIF - Université Paris Diderot

pCBPV *is a language well suited for writing* **probabilistic algorithm**. *It is equipped with the semantics of* **PCoh** *which is* **fully abstract** *thanks to* **quantitative** *properties.*

1. Probabilistic PCF and Pcoh
2. An implementation issue
3. Probabilistic Call by push Value and Pcoh

pPCF
~~pCBPV~~ *is a language well suited for writing* **probabilistic algorithm**. *It is equipped with the semantics of* **PCoh** *which is* **fully abstract** *thanks to* **quantitative** *properties.*

1. Probabilistic PCF and Pcoh
2. An implementation issue
3. Probabilistic Call by push Value and Pcoh

pPCF
~~**pCBPV**~~ *is a language* ~~*well suited*~~ *for writing* **probabilistic algorithm**. *It is equipped with the semantics of* **PCoh** *which is* **fully abstract** *thanks to* **quantitative** *properties.*

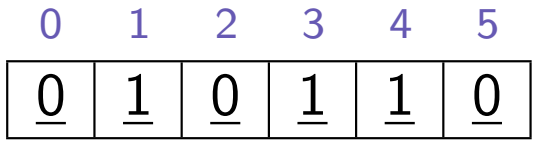
1. Probabilistic PCF and Pcoh
2. An implementation issue
3. Probabilistic Call by push Value and Pcoh

Probabilistic algorithm :

A Las Vegas example.

An example of Randomized algorithm

Input : A $\underline{0}/\underline{1}$ array of length $n \geq 2$ in which half cells are $\underline{0}$.



$f : 0, 2, 5 \mapsto \underline{0}, \quad 1, 3, 4 \mapsto \underline{1}$

Output : Find the index of a cell containing $\underline{0}$.

```
let rec LasVegas (f: nat -> nat) (n:nat) =
        let k = random n in
            if (f k = 0) then k
            else LasVegas f n
```

This algorithm succeeds with probability one.

- Success in 1 step is : $\frac{1}{2}$.
- Success in 2 steps is : $\frac{1}{2^2}$.
- Success in n steps is : $\frac{1}{2^n}$.

Success in any steps is :

$$\sum_{k=1}^{\infty} \frac{1}{2^k} = 1.$$

The Denotational Semantics Hammer

pPCF, Pcoh and Full abstraction

POPL'14 – "Probabilistic Coherence Spaces are Fully Abstract for Probabilistic PCF" with T. Ehrhard and M Pagani

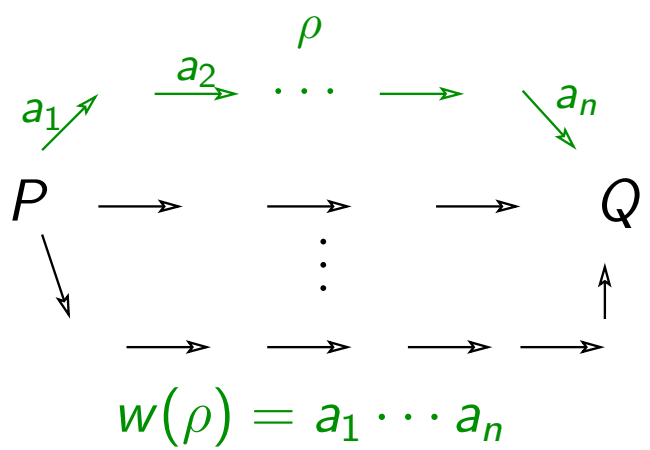
Types : $\sigma, \tau = \mathtt{nat} \mid \sigma \Rightarrow \tau$

Syntax :

$$\mathtt{N}, \mathtt{P}, \mathtt{Q} := \underline{\mathtt{n}} \mid \mathbf{pred}(\mathtt{N}) \mid \mathbf{succ}(\mathtt{N}) \mid \mathtt{x} \mid \lambda \mathtt{x}^{\sigma}\, \mathtt{P} \mid (\mathtt{P})\mathtt{Q} \mid \mathbf{fix}(\mathtt{M})$$
$$\mid \text{if } (N = \underline{0}) \text{ then } P \text{ else } Q \mid \mathbf{coin}(p), \text{ when } 0 \leq p \leq 1$$

Operational Semantics : $P \xrightarrow{\mathtt{a}} Q$

P reduces to Q in one step with probability a



$$\text{if } (\underline{0} = \underline{0}) \text{ then } P \text{ else } Q \xrightarrow{\mathtt{1}} P$$

$$\text{if } (\underline{n+1} = \underline{0}) \text{ then } P \text{ else } Q \xrightarrow{\mathtt{1}} Q$$

$$\mathbf{coin}(p) \xrightarrow{\mathtt{p}} \underline{0}$$

$$\mathbf{coin}(p) \xrightarrow{\mathtt{1-p}} \underline{1}$$

$$\mathbf{Proba}(P \xrightarrow{*} Q) = \sum_{\rho} w(\rho)$$

- 1999 – "Between Logic and Quantic : A tract". J. Y. Girard
- 2011 – "Probabilistic coherence spaces as a model of higher order probabilistic computation." V. Danos and T. Ehrhard

Probabilistic Coherent Spaces : $\mathcal{X} = (|\mathcal{X}|, \mathrm{P}(\mathcal{X}))$

where $|\mathcal{X}|$ is a countable set
and $\mathrm{P}(\mathcal{X}) \subseteq (\mathbb{R}^+)^{|\mathcal{X}|}$

To ensure PCoh to be a model, $\mathrm{P}(\mathcal{X})$ are subject to biorthogonality, covering and boundedness conditions.

Type Example : $\boxed{[\![\mathtt{nat}]\!] = (\mathbb{N}, \mathrm{P}(\mathtt{nat}) = \{(\lambda_n) \mid \sum_n \lambda_n \leq 1\})}$

Data Example : if $\mathtt{M} : \mathtt{nat}$, then $[\![\mathtt{M}]\!] \in \mathrm{P}(\mathtt{nat}) \subseteq (\mathbb{R}^+)^{\mathbb{N}}$
is a subprobability distributions.

Fair coin :

$$[\![\mathtt{coin}(\tfrac{1}{2})]\!] = \begin{array}{cccccc} & \overset{0}{\downarrow} & & \overset{1}{\downarrow} & \cdots & \cdots \\ (\tfrac{1}{2}, & & \tfrac{1}{2}, & & 0, & \ldots) \end{array}$$

Probabilistic coherent Maps : $f : (|\mathcal{X}|, \mathrm{P}(\mathcal{X})) \to (|\mathcal{Y}|, \mathrm{P}(\mathcal{Y}))$

defined as a **matrix** $M(f) \in (\mathbb{R}^+)^{\mathcal{M}_{\mathrm{fin}}(|\mathcal{X}|)\times|\mathcal{Y}|}$ with

$$f(x) = \sum_{[a_1,\ldots,a_n]\in\mathcal{M}_{\mathrm{fin}}(|\sigma|)} M(f)_{[a_1,\ldots,a_n]} \cdot \prod_{1\leq i\leq n} x_{a_i}$$

so that, the **analytic function** $f : (\mathbb{R}^+)^{|\mathcal{X}|} \to (\mathbb{R}^+)^{|\mathcal{Y}|}$ **preserves** probabilistic coherence : $f(\mathrm{P}(\mathcal{X})) \subseteq \mathrm{P}(\mathcal{Y})$

Example : if $\mathtt{P} : \mathtt{nat} \to \mathtt{nat}$, then $[\![\mathtt{P}]\!] : (\mathbb{R}^+)^{\mathbb{N}} \to (\mathbb{R}^+)^{\mathbb{N}}$ is an analytic function preserving subprobability distributions.

`Once : nat → nat`

Input : an integer `n`

Output : `if n=0 then 42 else rand 2`

$$\begin{array}{c} \begin{array}{cccc} \scriptstyle 0\downarrow & \scriptstyle 1\downarrow & \scriptstyle\cdots & \scriptstyle\cdots \end{array} \\ \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \cdots \\ 0 & \frac{1}{2} & \frac{1}{2} & \cdots \\ 0 & 0 & 0 & \cdots \\ \cdots & 0 & \cdots & \ddots \\ 1 & 0 & \cdots & \\ \cdots & 0 & \cdots & \ddots \end{pmatrix} \end{array} \begin{array}{l} \\ \to 0 \\ \to 1 \\ \vdots \\ \vdots \\ \to 42 \\ \vdots \end{array}$$

`Twice : nat → nat`

Input : an integer `n`

Output : `if n=0 then 42 else rand n`

$$\begin{array}{rcl} ([0], 42) & \mapsto & 1 \\ ([n_1, n_2], k) & \mapsto & \frac{1}{n_1+1} + \frac{1}{n_2+1} \\ & \text{if} & 0 \leq k \leq n_1 \leq n_2 \\ ([n_1, n_2], k) & \mapsto & \frac{1}{n_2+1} \\ & \text{if} & n_1 < k \leq n_2 \\ \text{Otherwise} & & 0 \end{array}$$

Probabilistic Data :

If $\mathtt{x}:\mathtt{nat}$, then $[\![\mathtt{x}]\!] = (\mathtt{x_n})_{\mathtt{n}\in\mathbb{N}}$

where x_n is the probability that $\mathtt{x}$ is n.

Probabilistic Program : $\mathtt{P} : \mathtt{nat} \to \mathtt{nat}$

where $[\![\mathtt{P\ x}]\!]_\mathtt{n}$ is the probability that $\mathtt{P\ x}$ computes n.

$$[\![\mathtt{Once}]\!] \in (\mathbb{R}^+)^{\mathbb{N}\times\mathbb{N}}$$

$$[\![\mathtt{Twice}]\!] \in (\mathbb{R}^+)^{\mathcal{M}_{\mathsf{fin}}(\mathbb{N})\times\mathbb{N}}$$

$$[\![\mathtt{Once\ x}]\!]_\mathtt{n} = [\![\mathtt{Once}]\!] \cdot [\![\mathtt{x}]\!] = \sum_k [\![\mathtt{Once}]\!]_{(\mathtt{k},\mathtt{n})}\, [\![\mathtt{x}]\!]_\mathtt{k}$$

$$[\![\mathtt{Twice\ x}]\!]_\mathtt{n} = [\![\mathtt{Twice}]\!] \cdot [\![\mathtt{x}]\!]^! = \sum_\mu [\![\mathtt{Twice}]\!]_{(\mu,\mathtt{n})} \prod_{\mathtt{k}\in\mu} [\![\mathtt{x}]\!]_\mathtt{k}$$

Operational semantics :

$\mathrm{Proba}(P \to^* v)$ is the probability that P computes v.

Denotational semantics :

Programs can be seen as **distribution transformers** which are **analytic functions**, so that

Adequacy Lemma :

If $\vdash M : \mathtt{nat}$, then $\forall n \in \mathbb{N}, [\![M]\!]_n = \mathrm{Proba}(M \to^* n)$

Probabilistic Full Abstraction :

Pcoh		**pPCF**
$[\![P]\!] = [\![Q]\!]$	Adequacy $\Rightarrow$ $\Leftarrow$ Full Completeness	$P \simeq_o Q$ $\forall C[\,], \forall v$ $\mathrm{Proba}(C[P] \to^* v) = \mathrm{Proba}(C[Q] \to^* v)$

How to encode the LasVegas Algorithm ?

Input : A $\underline{0}/\underline{1}$ array of length $n \geq 2$ in which half cells are $\underline{0}$.

0	1	2	3	4	5
$\underline{0}$	$\underline{1}$	$\underline{0}$	$\underline{1}$	$\underline{1}$	$\underline{0}$

$f : 0, 2, 5 \mapsto \underline{0}, \quad 1, 3, 4 \mapsto \underline{1}$

Output : Find the index of a cell containing $\underline{0}$.

Caml encoding :

```
let rec LasVegas (f: nat -> nat) (n:nat) =
                let k = random n in
                    if (f k = 0) then k
                    else LasVegas f n
```

pPCF encoding : $\texttt{rand}\ \underline{0} = \Omega$

$$\texttt{rand}\ \underline{\texttt{n}+1} = \texttt{if}\ (\texttt{coin}(\texttt{n}+1) = \underline{0})\ \texttt{then}\ \underline{\texttt{n}+1}\ \texttt{else}\ \texttt{rand}\ \underline{\texttt{n}}$$

$$\textbf{fix}\ (\lambda \texttt{LasVegas}^{(\texttt{nat}\Rightarrow\texttt{nat})\Rightarrow\texttt{nat}\Rightarrow\texttt{nat}}\ \lambda \texttt{f}^{\texttt{nat}\Rightarrow\texttt{nat}} \lambda \texttt{n}^{\texttt{nat}}\ (\lambda \texttt{k}^{\texttt{nat}}\ \texttt{if}\ (\texttt{f}\ \texttt{k} = \underline{0})\ \texttt{then}\ \texttt{k}\ \texttt{else}\ \texttt{LasVegas}\ \texttt{f}\ \texttt{n})\ \texttt{rand}\ \texttt{n}$$

Call By Name reflecting semantics :
In pPCF, data are not computed once for all (CBV) but whenever it appears in the program (CBN). If it is used several times, it is re-evaluated each time it appears.

Probabilistic Data as Random variables :
Each time a data is computed, it randomly gives a new value.

pPCF encoding :

0	1	2	3	4	5
$\underline{0}$	$\underline{1}$	$\underline{0}$	$\underline{1}$	$\underline{1}$	$\underline{0}$

$$\mathbf{fix}\,\big(\lambda \mathtt{LasVegas}^{(\mathtt{nat}\Rightarrow\mathtt{nat})\Rightarrow\mathtt{nat}\Rightarrow\mathtt{nat}}\ \lambda \mathtt{f}^{\mathtt{nat}\Rightarrow\mathtt{nat}}\lambda \mathtt{n}^{\mathtt{nat}}\ (\lambda \mathtt{k}^{\mathtt{nat}}\ \mathtt{if}\ (\mathtt{f}\ \mathtt{k} = \underline{0})\ \mathtt{then}\ \mathtt{k}\ \mathtt{else}\ \mathtt{LasVegas}\ \mathtt{f}\ \mathtt{n})(\mathtt{rand}\ \mathtt{n})$$

$$\rightarrow\ \mathbf{fix}\,\big(\lambda \mathtt{LasVegas}^{(\mathtt{nat}\Rightarrow\mathtt{nat})\Rightarrow\mathtt{nat}\Rightarrow\mathtt{nat}}\ \lambda \mathtt{f}^{\mathtt{nat}\Rightarrow\mathtt{nat}}\lambda \mathtt{n}^{\mathtt{nat}}\ (\lambda \mathtt{k}^{\mathtt{nat}}\ \mathtt{if}\ (\mathtt{f}\ (\mathtt{rand}\ \mathtt{n}) = \underline{0})\ \mathtt{then}\ \mathtt{rand}\ \mathtt{n}\ \mathtt{else}\ \mathtt{LasVegas}\ \mathtt{f}\ \mathtt{n})$$

Towards a solution ?

`Twice : nat → nat`

Input : an integer `n`

Output :
```
if n=0 then 42
else rand n
```

$$
\begin{array}{rcll}
([0], 42) & \mapsto & 1 & \\
([n_1, n_2], k) & \mapsto & \frac{1}{n_1+1} + \frac{1}{n_2+1} & \\
 & & \text{if} & 0 \leq k \leq n_1 \leq n_2 \\
([n_1, n_2], k) & \mapsto & \frac{1}{n_2+1} & \\
 & & \text{if} & n_1 < k \leq n_2 \\
\text{Otherwise} & & 0 &
\end{array}
$$

`Memo : nat → nat`

Input : an integer `n`

Output :
```
Let z = n in
if z=0 then 42
else rand z
```

$$
\begin{array}{rcll}
([0], 42) & \mapsto & 1 & \\
([n], k) & \mapsto & \frac{1}{n+1} & \\
 & & \text{if} & 0 < n \\
\text{Otherwise} & & 0 &
\end{array}
$$

Integer semantics in Pcoh :

$$[\![\mathtt{nat}]\!] = \left(\mathbb{N}, \mathrm{P}(\mathtt{nat}) = \{(\lambda_{\mathtt{n}}) \mid \sum_{\mathtt{n}} \lambda_{\mathtt{n}} \leq 1\}\right)$$

A coalgebric structure in the *linear* Pcoh :

- Contraction : $c^{\mathtt{nat}} : \mathtt{nat} \to \mathtt{nat} \otimes \mathtt{nat}$

$$c^{\mathtt{nat}}(\textstyle\sum_n x_n e_n) = \sum_n x_n e_n \otimes e_n \qquad c^{\mathtt{nat}}_{n,(p,q)} = \begin{cases} 1 & \text{if } n = p = q \\ 0 & \text{otherwise} \end{cases}$$

- Weakening : $w^{\mathtt{nat}} : \mathtt{nat} \to 1$

$$w^{\mathtt{nat}}(\textstyle\sum_n x_n e_n) = \sum_n x_n e_* \qquad w^{\mathtt{nat}}_{n,*} = 1, \quad \text{for any } n$$

Call by Value for nat and Call by Name pPCF

New conditional :

$$\frac{\Gamma \vdash N : \mathtt{nat} \qquad \Gamma \vdash P : \sigma \qquad \Gamma, z : \mathtt{nat} \vdash \mathtt{Q} : \sigma}{\Gamma \vdash \text{if } (N = \underline{0}) \text{ then } P \text{ else } [z]\, Q : \sigma}$$

$$\text{if } (\underline{0} = \underline{0}) \text{ then } P \text{ else } [z]\, Q \xrightarrow{1} P$$

$$\text{if } (\underline{n+1} = \underline{0}) \text{ then } P \text{ else } [z]\, Q \xrightarrow{1} Q[\underline{n}/z]$$

Memorization of the value of an expression of type nat.

$$\textbf{let } x = M \textbf{ in } N = \text{if } (M = \underline{0}) \text{ then } N[\underline{0}/x] \text{ else } [z]\, N[\textbf{succ}(z)/x]$$

New pPCF encoding :

$$\begin{aligned} \mathtt{LasVegas} = \textbf{fix} \big(\lambda\, \mathtt{LV}^{(\mathtt{nat}\Rightarrow\mathtt{nat})\Rightarrow\mathtt{nat}\Rightarrow\mathtt{nat}} \quad \lambda \mathtt{f}^{\mathtt{nat}\Rightarrow\mathtt{nat}} \quad \lambda \mathtt{n}^{\mathtt{nat}} \\ \textbf{let } \mathtt{k} = \mathtt{rand\ n} \textbf{ in} \\ \textbf{if } (\mathtt{f\ k,\ k,\ (LV)\ f\ n}) \end{aligned}$$

pCBPV and Pcoh

Convenient, Adequate and Fully abstract.


TLCA'99 – "Call By Push Value : A Subsuming Paradigm". P. B. Levy


ESOP'16 – "Call-By-Push-Value from a Linear Logic point of view". T. Ehrhard


2016 – "Probabilistic Call By Push Value" with T. Ehrhard

Main Ideas behind **pCBPV** and **Pcoh**

Linear Logic inspired :
Linear application on **positive** types, Exponential,...

Coalgebraic Structure :
For `nat` but also for other **positive** types (list, streams,. . .).

pCBPV :
Allows to combine CBV (for data) and CBN.

Values are particular terms of **positive** type which are :

- freely discardable and duplicable,
- interpreted as morphisms of coalgebras.

Syntax :

pCBPV – Probabilistic Call By Push Value

Types :

$$\textbf{(positive)} \quad \phi, \psi, \ldots := \mathbf{1} \mid !\sigma \mid \phi \otimes \psi \mid \phi \oplus \psi \mid \zeta \mid \mathsf{Fix}\,\zeta \cdot \phi$$

$$\textbf{(general)} \quad \sigma, \tau \ldots := \phi \mid \phi \multimap \sigma$$

Programs : (general type)

$$\begin{aligned} M, N \ldots := {} & x \mid () \mid M^{!} \mid (M, N) \mid \mathsf{in}_1 M \mid \mathsf{in}_2 M \\ & \mid \lambda x^{\phi}\, M \mid \langle M \rangle N \mid \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2) \\ & \mid \mathsf{pr}_1 M \mid \mathsf{pr}_2 M \mid \mathsf{der}(M) \mid \mathsf{fix}\, x^{!\sigma}\, M \\ & \mid \mathsf{fold}(M) \mid \mathsf{unfold}(M) \mid \mathsf{coin}(p),\ p \in [0,1] \cap \mathbb{Q} \end{aligned}$$

Typing context : $\mathcal{P} = (x_1 : \phi_1, \ldots, x_k : \phi_k)$

Values : (positive type)

$$V, W \ldots := x \mid () \mid M^{!} \mid (V, W) \mid \mathsf{in}_1 V \mid \mathsf{in}_2 V \mid \mathsf{fold}(V)\,.$$

Types : $\sigma \Rightarrow \tau = !\sigma \multimap \tau$

(positive) $\phi, \psi, \ldots := \mathbf{1} \mid !\sigma \mid \phi \otimes \psi \mid \phi \oplus \psi \mid \zeta \mid \mathsf{Fix}\,\zeta \cdot \phi$

(general) $\sigma, \tau \ldots := \phi \mid \phi \multimap \sigma$

Programs : $\lambda x^{\sigma}\, M = \lambda x^{!\sigma}\, M$ and $(M)N = \langle M^{!} \rangle N$

$$\begin{aligned} M, N \ldots := {} & x \mid () \mid M^{!} \mid (M, N) \mid \mathsf{in}_1 M \mid \mathsf{in}_2 M \\ & \mid \lambda x^{\phi}\, M \mid \langle M \rangle N \mid \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2) \\ & \mid \mathsf{pr}_1 M \mid \mathsf{pr}_2 M \mid \mathsf{der}(M) \mid \mathsf{fix}\, x^{!\sigma}\, M \\ & \mid \mathsf{fold}(M) \mid \mathsf{unfold}(M) \mid \mathsf{coin}(p),\ p \in [0,1] \cap \mathbb{Q} \end{aligned}$$

Typing context : $\mathcal{P} = (x_1 : \phi_1, \ldots, x_k : \phi_k)$

Values :

$$V, W \ldots := x \mid () \mid M^{!} \mid (V, W) \mid \mathsf{in}_1 V \mid \mathsf{in}_2 V \mid \mathsf{fold}(V)\,.$$

Types : $\mathtt{nat} = \mathsf{Fix}\,\zeta \cdot \mathbf{1} \oplus \zeta$ and $\sigma \Rightarrow \tau = !\sigma \multimap \tau$

(positive) $\phi, \psi, \ldots := \mathbf{1} \mid !\sigma \mid \phi \otimes \psi \mid \phi \oplus \psi \mid \zeta \mid \mathsf{Fix}\,\zeta \cdot \phi$

(general) $\sigma, \tau \ldots := \phi \mid \phi \multimap \sigma$

Programs : $\mathbf{succ}(M) = \mathsf{in}_2 M$

$$M, N \ldots := x \mid () \mid M^! \mid (M, N) \mid \mathsf{in}_1 M \mid \mathsf{in}_2 M$$
$$\mid \lambda x^{\phi}\, M \mid \langle M \rangle N \mid \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2)$$
$$\mid \mathsf{pr}_1 M \mid \mathsf{pr}_2 M \mid \mathsf{der}(M) \mid \mathsf{fix}\, x^{!\sigma}\, M$$
$$\mid \mathsf{fold}(M) \mid \mathsf{unfold}(M) \mid \mathsf{coin}(p),\ p \in [0,1] \cap \mathbb{Q}$$

Typing context : $\mathcal{P} = (x_1 : \phi_1, \ldots, x_k : \phi_k)$

Values : $\underline{0} = \mathsf{in}_1()$ and $\underline{n+1} = \mathsf{in}_2 \underline{n}$

$$V, W \ldots := x \mid () \mid M^! \mid (V, W) \mid \mathsf{in}_1 V \mid \mathsf{in}_2 V \mid \mathsf{fold}(V)\,.$$

Types : $\mathtt{nat} = \mathsf{Fix}\,\zeta \cdot \mathbf{1} \oplus \zeta$ and $\sigma \Rightarrow \tau = !\sigma \multimap \tau$

(positive) $\phi, \psi, \ldots := \mathbf{1} \mid !\sigma \mid \phi \otimes \psi \mid \phi \oplus \psi \mid \zeta \mid \mathsf{Fix}\,\zeta \cdot \phi$

(general) $\sigma, \tau \ldots := \phi \mid \phi \multimap \sigma$

Programs : $\mathbf{pred}(M) = \mathsf{case}(M, x \cdot \underline{0}, z \cdot z)$

$$M, N \ldots := x \mid () \mid M^! \mid (M, N) \mid \mathsf{in}_1 M \mid \mathsf{in}_2 M$$
$$\mid \lambda x^\phi\, M \mid \langle M \rangle N \mid \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2)$$
$$\mid \mathsf{pr}_1 M \mid \mathsf{pr}_2 M \mid \mathsf{der}(M) \mid \mathsf{fix}\, x^{!\sigma}\, M$$
$$\mid \mathsf{fold}(M) \mid \mathsf{unfold}(M) \mid \mathsf{coin}(p),\ p \in [0,1] \cap \mathbb{Q}$$

Typing context : $\mathcal{P} = (x_1 : \phi_1, \ldots, x_k : \phi_k)$

Values : $\underline{0} = \mathsf{in}_1()$ and $\underline{n+1} = \mathsf{in}_2 \underline{n}$

$$V, W \ldots := x \mid () \mid M^! \mid (V, W) \mid \mathsf{in}_1 V \mid \mathsf{in}_2 V \mid \mathsf{fold}(V)\,.$$

Types : $\mathtt{nat} = \mathsf{Fix}\,\zeta \cdot \mathbf{1} \oplus \zeta$ and $\sigma \Rightarrow \tau = !\sigma \multimap \tau$

(positive) $\phi, \psi, \ldots := \mathbf{1} \mid !\sigma \mid \phi \otimes \psi \mid \phi \oplus \psi \mid \zeta \mid \mathsf{Fix}\,\zeta \cdot \phi$

(general) $\sigma, \tau \ldots := \phi \mid \phi \multimap \sigma$

Programs : $\mathsf{if}\,(M = \underline{0})\,\mathsf{then}\,P\,\mathsf{else}\,[z]\,Q = \mathsf{case}(M, x \cdot P, z \cdot Q)$

$$M, N \ldots := x \mid () \mid M^! \mid (M, N) \mid \mathsf{in}_1 M \mid \mathsf{in}_2 M$$
$$\mid \lambda x^{\phi}\, M \mid \langle M \rangle N \mid \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2)$$
$$\mid \mathsf{pr}_1 M \mid \mathsf{pr}_2 M \mid \mathsf{der}(M) \mid \mathsf{fix}\, x^{!\sigma}\, M$$
$$\mid \mathsf{fold}(M) \mid \mathsf{unfold}(M) \mid \mathsf{coin}(p),\ p \in [0,1] \cap \mathbb{Q}$$

Typing context : $\mathcal{P} = (x_1 : \phi_1, \ldots, x_k : \phi_k)$

Values : $\underline{0} = \mathsf{in}_1()$ and $\underline{n+1} = \mathsf{in}_2 \underline{n}$

$$V, W \ldots := x \mid () \mid M^! \mid (V, W) \mid \mathsf{in}_1 V \mid \mathsf{in}_2 V \mid \mathsf{fold}(V)\,.$$

Probabilistic Coherent Spaces (Pcoh) :

Linearity, Exponential and Coalgebras

LL-based denotational semantics :
Smcc ($\otimes$, $\multimap$,...), $*$-autonomous ($\bot$), Cartesian ($\parr$), Exponential comonad ($!\_$) with a strong symmetric monoidal structure

Fixpoints for formulae : (Types)

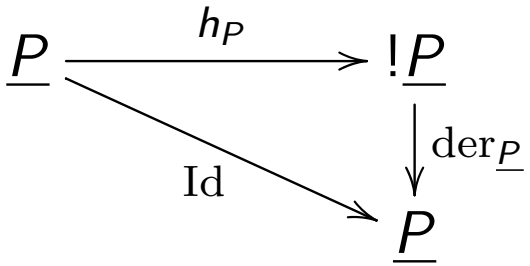
$$X \subseteq Y \text{ iff } |X| \subseteq |Y| \text{ and } \begin{cases} \text{if } u \in \mathrm{P}(X), \text{ then } \mathrm{ext}_{|Y|}(u) \in \mathrm{P}(Y) \\ \text{if } v \in \mathrm{P}(Y), \text{ then } \mathrm{res}_{|X|}(v) \in \mathrm{P}(X) \end{cases}$$

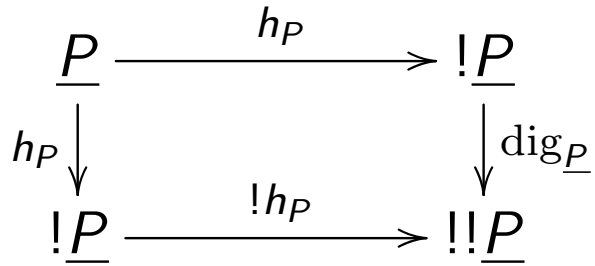
(**Pcoh**, $\subseteq$) is a cpo and all connectives are Scott Continuous.

Fixpoints for morphisms : (Programs)
if $f \in \mathbf{Pcoh}(!Y \otimes !X, X)$, then there is $f^\dagger \in \mathbf{Pcoh}(!Y, X)$ as morphisms, seen as analytic functions are Scott Continuous.

The category of coalgebras : **Pcoh**$^!$

$P = (\underline{P}, h_P)$ with $\underline{P} \in \mathbf{Pcoh}$ and $h_P \in \mathbf{Pcoh}(\underline{P}, !\underline{P})$ satisfies :





Positive types are coalgebras $[\![\phi]\!]^! \in \mathbf{Pcoh}^!$

$(!X, \mathrm{dig}_X) \in \mathbf{Pcoh}^!$ and coalgebras are stable by $\otimes$, $\oplus$, fixpoints.

Values are morphisms of coalgebras $[\![V]\!] \in \mathbf{Pcoh}^!([\![\mathcal{P}]\!]^!, [\![\phi]\!]^!)$

where V is a value and $\mathcal{P} \vdash V : \phi$ where $\mathcal{P} = (x_1 : \phi_1, \ldots, x_k : \phi_k)$ and $[\![\mathcal{P}]\!]^! = [\![\phi_1]\!]^! \otimes \cdots \otimes [\![\phi_k]\!]^!$.

A Cartesian Category : (not closed)

$(\mathbf{Pcoh}^!, \otimes, 1)$ is cartesian and for any coalgebra P, there is a contraction : $c_P : P \to P \otimes P$ and a weakening $w_P : P \to 1$

Linearization :

A positive type has structural rules, a function of type $\phi \multimap \sigma$ has no linearity restriction on the use of its argument.

Dense Coalgebras : if ϕ is positif, then $[\![\phi]\!]^!$ is dense.

$P = (\underline{P}, h_P)$ is **dense** if coalgebric points characterize morphisms :

$$\forall X \in \mathbf{Pcoh} \text{ and } \forall t, t' \in \mathbf{Pcoh}(\underline{P}, X),$$
$$\text{if } \forall v \in \mathbf{Pcoh}^!(1, P),\ t\, v = t'\, v, \text{ then } \forall u \in \mathbf{Pcoh}(1, \underline{P}),\ t\, u = t'\, u.$$

Already known for $t, t' \in \mathbf{Pcoh}(X, Y)$ as $t\, u^! = t'\, u^!$ implies $t = t'$.

Interpretation of **CBPV** in **Pcoh**

Positive types : $[\![\phi]\!]^! \in \mathbf{Pcoh}^!$

General types : $[\![\sigma]\!] \in \mathbf{Pcoh}$ with $[\![\phi]\!] = \underline{[\![\phi]\!]^!}$

Values : if $\mathcal{P} \vdash V : \phi$, then $[\![V]\!] \in \mathbf{Pcoh}^!([\![\mathcal{P}]\!]^!, [\![\phi]\!]^!)$.

Terms : if $\mathcal{P} \vdash M : \sigma$, then $[\![M]\!] \in \mathbf{Pcoh}([\![\mathcal{P}]\!], [\![\sigma]\!])$.

Probabilistic Full Abstraction :

The Adequacy Lemma

Let $P, Q : \sigma \quad \forall \alpha \in |\sigma|, \llbracket P \rrbracket_\alpha = \llbracket Q \rrbracket_\alpha$

Adequacy $\Downarrow$ $\Uparrow$ **Full Abstraction**

$$\forall C :!\sigma \multimap 1,$$
$$\mathsf{Proba}(\langle C \rangle P^! \xrightarrow{*} ()) = \mathsf{Proba}(\langle C \rangle Q^! \xrightarrow{*} ())$$

Adequacy Lemma :

If M is closed and $\vdash M : 1$, then $[\![M]\!] = \text{Proba}(M \to^* ())$

Ingredients : Tricky point : $[\![M]\!] \leq \text{Proba}(M \to^* ())$

- Logical relation between terms and elements of the model
- Pitt's Technique for recursive types with positive and negative occurences of type variables (fixpoint of tuple of relations)
- Two kinds of relations for positive and general types
- Hidden Biorthogonality closure for positive types
- Hidden Step Indexed Logical relation techniques (fold/unfold)

Adequacy proof :

If $[\![P]\!] = [\![Q]\!]$ then, $\text{Proba}(\langle C \rangle P \xrightarrow{*} ()) = \text{Proba}(\langle C \rangle Q \xrightarrow{*} ())$

1. Apply Adequacy Lemma : $\text{Proba}(\langle C \rangle P \xrightarrow{*} ()) = [\![\langle C \rangle P]\!]$.
2. Apply Compositionality :

$$[\![\langle C \rangle P]\!] = \sum_{\mu} [\![C]\!]_{\mu} \prod_{\alpha \in \mu} [\![P]\!]_{\alpha}^{\mu(\alpha)} = \sum_{\mu} [\![C]\!]_{\mu} \prod_{\alpha \in \mu} [\![Q]\!]_{\alpha}^{\mu(\alpha)} = [\![\langle C \rangle Q]\!]$$

Probabilistic Full Abstraction :

The Full Abstraction theorem

Let $P, Q : \sigma \quad \forall \alpha \in |\sigma|, [\![P]\!]_\alpha = [\![Q]\!]_\alpha$

Adequacy $\Downarrow \Uparrow$ **Full Abstraction**

$$\forall C :!\sigma \multimap 1,$$
$$\mathrm{Proba}(\langle C \rangle P^! \xrightarrow{*} ()) = \mathrm{Proba}(\langle C \rangle Q^! \xrightarrow{*} ())$$

Full Abstraction proof :

1. By **contradiction** : $\exists \alpha \in |\sigma|\,,\ [\![P]\!]_\alpha \neq [\![Q]\!]_\alpha$
2. Find **testing context** : T_α such that $[\![(T_\alpha)P]\!] \neq [\![(T_\alpha)Q]\!]$ (context only depends on T_α)
3. Prove **definability** : $T_\alpha \in PPCF$ (uses coin(p) and analyticity)
4. Apply **Adequacy Lemma** : Proba$(\langle T_\alpha \rangle P^! \xrightarrow{*} ()) \neq$ Proba$(\langle T_\alpha \rangle Q^! \xrightarrow{*} ())$.

Tricky Hidden part :

- Defining contexts for positive and general types
- Definability relies on dense coalgebras

To sum up :

pCBPV *is a language well suited for writing* **probabilistic algorithm**, *combining CBN and CBV. It is equipped with the semantics of* **PCoh** *which is* **fully abstract** *thanks to* **quantitative** *properties.*

Further directions :

- use this language to study various effects (non-determinism, states etc) using computational monads on the linear category.
- resource calculi for CBPV, Taylor expansion