

Probabilistic Full abstraction

joint works with **Thomas Ehrhard** and **Michele Pagani**

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 0/1 array of length $n \geq 2$ in which half cells are 0.

0	1	2	3	4	5
<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>

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

Output : Find the index of a cell containing 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

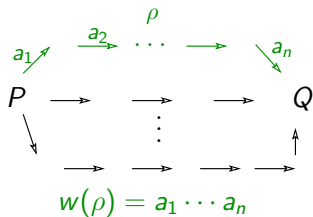
Types : $\sigma, \tau = \text{nat} \mid \sigma \Rightarrow \tau$

Syntax :

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

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

P reduces to Q in one step with probability a





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

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

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

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

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

-  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 :

$$\mathcal{X} = (|\mathcal{X}|, P(\mathcal{X}))$$

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

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

Type Example :

$$\llbracket \text{nat} \rrbracket = (\mathbb{N}, P(\text{nat}) = \{(\lambda_n) \mid \sum_n \lambda_n \leq 1\})$$

Data Example :

if $M : \text{nat}$, then $\llbracket M \rrbracket \in P(\text{nat}) \subseteq (\mathbb{R}^+)^{\mathbb{N}}$
is a subprobability distributions.

Fair coin :

$$\llbracket \text{coin}(\frac{1}{2}) \rrbracket = \left(\begin{array}{cccc} & 0 & 1 & \dots & \dots \\ & \downarrow & \downarrow & & \\ \frac{1}{2}, & \frac{1}{2}, & 0, & \dots \end{array} \right)$$

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

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

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

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

Example : if $P : \text{nat} \rightarrow \text{nat}$, then $\llbracket P \rrbracket : (\mathbb{R}^+)^{\text{N}} \rightarrow (\mathbb{R}^+)^{\text{N}}$ is an analytic function preserving subprobability distributions.

The semantics of Probabilistic Programs

Once : $\text{nat} \rightarrow \text{nat}$

Input : an integer n

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

$$\begin{array}{cccc}
 0 & 1 & \dots & \dots \\
 \downarrow & \downarrow & & \\
 \left(\begin{array}{cccc}
 0 & \frac{1}{2} & \frac{1}{2} & \dots \\
 0 & \frac{1}{2} & \frac{1}{2} & \dots \\
 0 & 0 & 0 & \dots \\
 \dots & 0 & \dots & \ddots \\
 1 & 0 & \dots & \\
 \dots & 0 & \dots & \ddots
 \end{array} \right) & \begin{array}{l} \rightarrow 0 \\ \rightarrow 1 \\ \vdots \\ \vdots \\ \rightarrow 42 \\ \vdots \end{array}
 \end{array}$$

Twice : $\text{nat} \rightarrow \text{nat}$

Input : an integer n

Output : if $n=0$ then 42
 else n

$$\begin{array}{ll}
 ([0], 42) & \mapsto 1 \\
 ([n_1, n_2], n_2) & \mapsto 1 \\
 & \text{if } n_1 \neq 0 \\
 \text{Otherwise} & 0
 \end{array}$$

Probabilistic Data :

If $\mathbf{x} : \text{nat}$, then $\llbracket \mathbf{x} \rrbracket = (x_n)_{n \in \mathbb{N}}$

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

Probabilistic Program : $P : \text{nat} \rightarrow \text{nat}$

where $\llbracket P \ \mathbf{x} \rrbracket_n$ is the probability that $P \ \mathbf{x}$ computes n .

$$\llbracket \text{Once} \rrbracket \in (\mathbb{R}^+)^{\mathbb{N} \times \mathbb{N}}$$

$$\llbracket \text{Twice} \rrbracket \in (\mathbb{R}^+)^{\mathcal{M}_{\text{fin}}(\mathbb{N}) \times \mathbb{N}}$$

$$\begin{aligned} \llbracket \text{Once } \mathbf{x} \rrbracket_n &= \llbracket \text{Once} \rrbracket \cdot \llbracket \mathbf{x} \rrbracket \\ &= \sum_k \llbracket \text{Once} \rrbracket_{(k,n)} \llbracket \mathbf{x} \rrbracket_k \end{aligned}$$

$$\begin{aligned} \llbracket \text{Twice } \mathbf{x} \rrbracket_n &= \llbracket \text{Twice} \rrbracket \cdot \llbracket \mathbf{x} \rrbracket^! \\ &= \sum_{\mu} \llbracket \text{Twice} \rrbracket_{(\mu,n)} \prod_{k \in \mu} \llbracket \mathbf{x} \rrbracket_k \end{aligned}$$

Operational semantics :

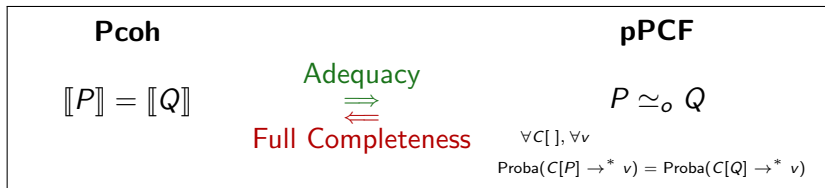
$\text{Proba}(P \rightarrow^* v)$ is the probability that P computes v .

Denotational semantics :

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

Adequacy Lemma :

If $\vdash M : \text{nat}$, then $\forall n \in \mathbb{N}, \llbracket M \rrbracket_n = \text{Proba}(M \rightarrow^* n)$

Probabilistic Full Abstraction :

How to encode the LasVegas Algorithm?

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

0	1	2	3	4	5
<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>

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

Output : Find the index of a cell containing 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 : $\text{rand } \underline{0} = \Omega$

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

```
fix ( $\lambda \text{LasVegas}^{(\text{nat} \Rightarrow \text{nat}) \Rightarrow \text{nat} \Rightarrow \text{nat}} \lambda f^{\text{nat} \Rightarrow \text{nat}} \lambda n^{\text{nat}}$ 
  ( $\lambda k^{\text{nat}}$  if (f k = 0) then k
  else LasVegas f n) rand 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
<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>

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

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

Twice : $\text{nat} \rightarrow \text{nat}$

Input : an integer n

Output : if $n=0$ then 42
 else n

$([0], 42) \mapsto 1$

$([n_1, n_2], n_2) \mapsto 1$

if $n_1 \neq 0$

Otherwise 0

Memo : $\text{nat} \rightarrow \text{nat}$

Input : an integer n

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

$([0], 42) \mapsto 1$

$([n], n) \mapsto 1$

if $n \neq 0$

Otherwise 0

Integer semantics in Pcoh :

$$\llbracket \text{nat} \rrbracket = \left(\mathbb{N}, P(\text{nat}) = \{(\lambda_n) \mid \sum_n \lambda_n \leq 1\} \right)$$

A coalgebraic structure in the *linear* Pcoh :

- Contraction : $c^{\text{nat}} : \text{nat} \rightarrow \text{nat} \otimes \text{nat}$

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

- Weakening : $w^{\text{nat}} : \text{nat} \rightarrow 1$

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

- 2015 – "Probabilistic Coherence Spaces are Fully Abstract for Probabilistic PCF" with T. Ehrhard and M. Pagani

New conditional :

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

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

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

Memorization of the value of an expression of type nat.




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

New pPCF encoding :

LasVegas = **fix** $(\lambda LV^{(\text{nat} \Rightarrow \text{nat}) \Rightarrow \text{nat} \Rightarrow \text{nat}} \lambda f^{\text{nat} \Rightarrow \text{nat}} \lambda n^{\text{nat}}$
 $\quad \mathbf{let} \ k = \mathbf{rand} \ n \ \mathbf{in}$
 $\quad \mathbf{if} (f \ k = 0) \ \mathbf{then} \ k \ \mathbf{else} \ (LV) \ f \ n$

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

Linear Logic inspired :

Linear application on **positive** types, Exponential,...

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.

Coalgebraic Structure :

For the interpretation of **positive** types (nat, list, streams, ...).

Syntax :

pCBPV – Probabilistic Call By Push Value

Types :

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

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

Values : (positive type)

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

Programs : (general type)

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

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

Types :

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

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

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

Values :

$V, W \dots := x \mid () \mid M^\dagger \mid (V, W) \mid \text{in}_1 V \mid \text{in}_2 V \mid \text{fold}(V).$

Programs :

$$\lambda x^\sigma M = \lambda x^{!\sigma} M \text{ and } (M)N = \langle M \rangle N^\dagger$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Some glimpse of **pCBPV** operational semantics.

Weak Reduction for Values.

$$\frac{}{\text{der}(M!) \xrightarrow{w} M} \quad \frac{}{\langle \lambda x^\phi M \rangle V \xrightarrow{w} M[V/x]}$$
$$\frac{}{\text{case}(\text{in}_i V, x_1 \cdot M_1, x_2 \cdot M_2) \xrightarrow{w} M_i[V/x_i]}$$

Probabilistic Reduction.

$$\frac{M \xrightarrow{w} M'}{M \xrightarrow{1} M'} \quad \frac{}{\text{coin}(p) \xrightarrow{p} \underline{0} = \text{in}_1()} \quad \frac{}{\text{coin}(p) \xrightarrow{1-p} \underline{1} = \text{in}_2(\text{in}_1())}$$
$$\frac{M \xrightarrow{p} M'}{\text{der}(M) \xrightarrow{p} \text{der}(M')} \quad \frac{M \xrightarrow{p} M'}{\langle M \rangle V \xrightarrow{p} \langle M' \rangle V} \quad \frac{N \xrightarrow{p} N'}{\langle M \rangle N \xrightarrow{p} \langle M \rangle N'}$$
$$\frac{M \xrightarrow{p} M'}{\text{case}(M, x_1 \cdot M_1, x_2 \cdot M_2) \xrightarrow{p} \text{case}(M', x_1 \cdot M_1, x_2 \cdot M_2)}$$

Some glimpse of **pCBPV** operational semantics.

Weak Reduction for Values.

$$\frac{}{\text{der}(M!) \xrightarrow{w} M} \quad \frac{}{\langle \lambda x^\phi M \rangle V \xrightarrow{w} M[V/x]}$$
$$\frac{}{\text{case}(\text{in}_i V, x_1 \cdot M_1, x_2 \cdot M_2) \xrightarrow{w} M_i[V/x_i]}$$

Probabilistic Reduction.

$$\frac{M \xrightarrow{w} M'}{M \xrightarrow{1} M'} \quad \frac{}{\text{coin}(p) \xrightarrow{p} \underline{0} = \text{in}_1()} \quad \frac{}{\text{coin}(p) \xrightarrow{1-p} \underline{1} = \text{in}_2(\text{in}_1())}$$
$$\frac{M \xrightarrow{p} M'}{\text{der}(M) \xrightarrow{p} \text{der}(M')} \quad \frac{M \xrightarrow{p} M'}{\langle M \rangle V \xrightarrow{p} \langle M' \rangle V} \quad \frac{N \xrightarrow{p} N'}{\langle M \rangle N \xrightarrow{p} \langle M \rangle N'}$$
$$\frac{M \xrightarrow{p} M'}{\text{case}(M, x_1 \cdot M_1, x_2 \cdot M_2) \xrightarrow{p} \text{case}(M', x_1 \cdot M_1, x_2 \cdot M_2)}$$

Some glimpse of **pCBPV** operational semantics.

Weak Reduction for Values.

$$\frac{}{\text{der}(M!) \xrightarrow{w} M} \quad \frac{}{\langle \lambda x^\phi M \rangle V \xrightarrow{w} M[V/x]}$$
$$\frac{}{\text{case}(in_i V, x_1 \cdot M_1, x_2 \cdot M_2) \xrightarrow{w} M_i[V/x_i]}$$

Probabilistic Reduction.

$$\frac{M \xrightarrow{w} M'}{M \xrightarrow{1} M'} \quad \frac{}{\text{coin}(p) \xrightarrow{p} \underline{0} = \text{in}_1()} \quad \frac{}{\text{coin}(p) \xrightarrow{1-p} \underline{1} = \text{in}_2(\text{in}_1())}$$
$$\frac{M \xrightarrow{p} M'}{\text{der}(M) \xrightarrow{p} \text{der}(M')} \quad \frac{M \xrightarrow{p} M'}{\langle M \rangle V \xrightarrow{p} \langle M' \rangle V} \quad \frac{N \xrightarrow{p} N'}{\langle M \rangle N \xrightarrow{p} \langle M \rangle N'}$$
$$\frac{M \xrightarrow{p} M'}{\text{case}(M, x_1 \cdot M_1, x_2 \cdot M_2) \xrightarrow{p} \text{case}(M', x_1 \cdot M_1, x_2 \cdot M_2)}$$

Probabilistic Coherent Spaces (Pcoh) :

Linearity, Exponential and Coalgebras

LL-based denotational semantics :

Smcc ($\otimes, \multimap, \dots$), *-autonomous (\perp), Cartesian (\wp), Exponential comonad ($!_-$) with a strong symmetric monoidal structure

Fixpoints for formulae : (Types)

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

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

Fixpoints for morphisms : (Programs)

The fixpoint operator can be interpreted as analytic functions are in particular Scott Continuous.

The category of coalgebras : $\mathbf{Pcoh}^!$

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

$$\begin{array}{ccc} \underline{P} & \xrightarrow{h_P} & !\underline{P} \\ & \searrow \text{Id} & \downarrow \text{der}_{\underline{P}} \\ & & \underline{P} \end{array}$$

$$\begin{array}{ccc} \underline{P} & \xrightarrow{h_P} & !\underline{P} \\ h_P \downarrow & & \downarrow \text{dig}_{\underline{P}} \\ !\underline{P} & \xrightarrow{!h_P} & !!\underline{P} \end{array}$$

Positive types are coalgebras :

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

$(!X, \text{dig}_X) \in \mathbf{Pcoh}^!$ 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 $\mathcal{P} = (x_1 : \phi_1, \dots, x_k : \phi_k)$ and $[[\mathcal{P}]]^! = [[\phi_1]]^! \otimes \dots \otimes [[\phi_k]]^!$.

A Cartesian Category : $(\mathbf{Pcoh}^!, \otimes, 1)$ is cartesian (not closed)

Linearization : Positive types have structural rules.

Contraction $c_P : P \rightarrow P \otimes P$ and weakening $w_P : P \rightarrow 1$.

No linearity restriction on arguments of terms of type $\phi \multimap \sigma$.

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

$P = (\underline{P}, h_P)$ is **dense** if coalgebraic 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 if $\forall u \in \mathbf{Pcoh}(1, X), t u^! = t' u^!$ implies $t = t'$.

Positive types :

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

General types :

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

Values :

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

Terms :

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

Call-by-Value :

Values are of positive types. They can be duplicated and erased, accordingly to their semantics (coalgebraic structure).

Probabilistic algorithm :

It is now possible to encode program algorithms that handle base types such as lists or streams.

Full-abstraction :

Pcoh is a semantics of pCBPV which enjoys good properties.

Probabilistic Full Abstraction :

The Adequacy Lemma

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

Adequacy $\Downarrow \Uparrow$ Full Abstraction

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

Adequacy Lemma :

If M is closed and $\vdash M : 1$, then $\llbracket M \rrbracket = \text{Proba}(M \rightarrow^* ())$

Ingredients : Tricky point : $\llbracket M \rrbracket \leq \text{Proba}(M \rightarrow^* ())$

- * Logical relation between terms and elements of the model
- * Pitt's Technique for recursive types with positive and negative occurrences 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 $\llbracket P \rrbracket = \llbracket Q \rrbracket$ then, $\text{Proba}(\langle C \rangle P^! \xrightarrow{*} ()) = \text{Proba}(\langle C \rangle Q^! \xrightarrow{*} ())$

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

$$\llbracket \langle C \rangle P^! \rrbracket = \sum_{\mu} \llbracket C \rrbracket_{\mu} \prod_{\alpha \in \mu} \llbracket P \rrbracket_{\alpha}^{\mu(\alpha)} = \sum_{\mu} \llbracket C \rrbracket_{\mu} \prod_{\alpha \in \mu} \llbracket Q \rrbracket_{\alpha}^{\mu(\alpha)} = \llbracket \langle C \rangle Q^! \rrbracket$$

Probabilistic Full Abstraction :

The Full Abstraction theorem

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

Adequacy $\Downarrow \Uparrow$ Full Abstraction

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

Full Abstraction proof :

- 1 By **contradiction** : $\exists \alpha \in |\sigma|, \llbracket P \rrbracket_\alpha \neq \llbracket Q \rrbracket_\alpha$
- 2 Find **testing context** : T_α such that $\llbracket \langle T_\alpha \rangle P^! \rrbracket \neq \llbracket \langle T_\alpha \rangle Q^! \rrbracket$
(context only depends on α)
- 3 Prove **definability** : $T_\alpha \in \mathbf{pCBPV}$ (uses $\text{coin}(p)$).
- 4 Apply **Adequacy Lemma** :
 $\text{Proba}(\langle T_\alpha \rangle P^! \xrightarrow{*} ()) \neq \text{Proba}(\langle T_\alpha \rangle Q^! \xrightarrow{*} ())$.

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
- What about reactive programming ?