WIP 2013
15-18 December 2013

A Quantitative Path
between Syntax and Semantics

joint works with
Thomas Ehrhard, **Marie Kerjean** and **Michele Pagani**

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

Laboratoire PPS - Université Paris Diderot

What is it ?

- Linear Logic approach to semantics [Girard]
- Structured Vector spaces, Linear maps and Entire functions

What is it used for ?

- *PCoh Spaces are fully abstract for Probabilist PCF*
  POPL14 with T. Ehrhard and M. Pagani
- *A convenient model of lambda calculus*
  Master dissertation of M. Kerjean

Postdoc hiring with M. Pagani

- COmputing with QUAntitative Semantics [CoQuaS]
- `http://lipn.univ-paris13.fr/~pagani/pmwiki/pmwiki.php/Coquas/Coquas`

Quantitative Semantics:

What is it ?

Intuitions from Probabilistic Computation

Modeling Probabilistic Computation:

Type: set of positive vectors

Program: function seen as a positive matrix

Interaction: composition seen as multiplication

Example: `nat`

`Coin:nat` outcomes the toss of a fair coin.

`Random n:nat` outcomes uniformly any $\{0,\ldots,n-1\}$.

Ground Type Programs as Vectors: **Random Variables**.

$$[\![\texttt{Coin}]\!] = \big(\underset{0}{\underset{\downarrow}{\tfrac{1}{2}}}, \underset{1}{\underset{\downarrow}{\tfrac{1}{2}}}, \underset{2}{\underset{\downarrow}{0}}, \underset{\ldots}{\ldots}\big) \quad \text{and} \quad [\![\texttt{Random n}]\!] = \big(\underset{0}{\underset{\downarrow}{\tfrac{1}{n}}}, \underset{\ldots}{\ldots}, \underset{n-1}{\underset{\downarrow}{\tfrac{1}{n}}}, 0, \underset{\ldots}{\ldots}\big)$$

outcomes:

Subprobability Distributions over $\mathbb{N}$: $[\![\texttt{nat}]\!] \subseteq (\mathbb{R}^+)^{\mathbb{N}}$.

$$\boxed{[\![\texttt{nat}]\!] = \left\{(\lambda_n)_{n\in\mathbb{N}} \;\middle|\; \forall n, \lambda_n \in \mathbb{R}^+ \text{ and } \sum_n \lambda_n \leq 1\right\}}$$

Example: $\texttt{Random} : \texttt{nat} \to \texttt{nat}$

Input: an integer n

Output: any integer $\{0, \ldots, n-1\}$ uniformly chosen.

Higher Type Programs as Matrices: $[\![\texttt{Random}]\!] \in (\mathbb{R}^+)^{(\mathbb{N}\times\mathbb{N})}$.

$$[\![\texttt{Random}]\!] = \begin{array}{c c} \begin{array}{cccccc} \text{inputs: } 0\downarrow & 1\downarrow & 2\downarrow & \cdots & n\downarrow & \cdots \end{array} & \text{outputs} \\ \begin{pmatrix} 0 & 1 & \frac{1}{2} & \cdots & \frac{1}{n} & \cdots \\ 0 & 0 & \frac{1}{2} & \cdots & \frac{1}{n} & \cdots \\ \vdots & \vdots & 0 & \ddots & \vdots & \\ & & \vdots & 0 & \frac{1}{n} & \\ & & & \vdots & \ddots & \end{pmatrix} & \begin{array}{c} \to 0 \\ \to 1 \\ \vdots \\ \to n-1 \\ \vdots \end{array} \end{array}$$

Functions preserving subprobability distribution.

`Once` : `nat` $\to$ `nat`

Input: an integer `x`

Output:
```
if x=0
  then Coin
  else 42
```

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

$([0],\ 0) \mapsto 1/2$

$([0],\ 1) \mapsto 1/2$

$([a],\ 42) \mapsto 1$ if $a \neq 0$

$(m,\ k) \mapsto 0$ otherwise.

`Twice` : `nat` $\to$ `nat`

Input: an integer `x`

Output:
```
if x=0
  then (if x=0
          then Coin
          else 42)
  else (if x=0
          then 42
          else 0)
```

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

$([0,0],\ 0) \mapsto 1/2$

$([0,0],\ 1) \mapsto 1/2$

$([0,a],\ 42) \mapsto 2$ if $a \neq 0$

$([a,b],\ 0) \mapsto 1$ if $a,b \neq 0$

$(m,\ a) \mapsto 0$ otherwise.

Probabilistic Modeling of **Interaction**

Reminder: For $\texttt{x:nat}$, $a \in \mathbb{N}$

$[\![\texttt{x}]\!]_a =$ probability that random variable $\texttt{x}$ outcomes a.

Decompose computation as disjoint events:

$m = [a_1, \ldots, a_k]$ gathers effectively used input values.

$$[\![\texttt{P x}]\!]_{\texttt{b}} = \sum_{\texttt{m}} [\![\texttt{P}]\!]_{(\texttt{m},\texttt{b})} \cdot [\![\texttt{x}]\!]_{\texttt{a}_1} \cdots [\![\texttt{x}]\!]_{\texttt{a}_\texttt{k}}$$

Linear Programs as **Linear Functions**

$$[\![\texttt{Once x}]\!]_{\texttt{b}} = \sum_{\texttt{a}} [\![\texttt{Once}]\!]_{([\texttt{a}],\texttt{b})} \, [\![\texttt{x}]\!]_{\texttt{a}}$$

Non Linear Program as **Entire Functions**

$$[\![\texttt{Twice x}]\!]_{\texttt{b}} = \sum_{\texttt{a},\texttt{a}'} [\![\texttt{Twice}]\!]_{([\texttt{a},\texttt{a}'],\texttt{b})} \, [\![\texttt{x}]\!]_{\texttt{a}} \, [\![\texttt{x}]\!]_{\texttt{a}'}$$

Type: **Module**

Set of admissible vectors/Topological structure

$[\![\texttt{nat}]\!] = \{(\lambda_{\texttt{a}})_{\texttt{a}\in\mathbb{N}} \mid \forall \texttt{a}, \lambda_{\texttt{a}} \in \mathbb{R}^+ \text{ and } \sum_{\texttt{a}} \lambda_{\texttt{a}} \leq 1\} \subseteq (\mathbb{R})^{\mathbb{N}}$

Linear Program: **Linear Functions**

Preserving the additional structure

$[\![\texttt{Once}]\!] : \texttt{x} \mapsto \sum_{\texttt{a}} [\![\texttt{Once}]\!]_{([\texttt{a}],\texttt{b})} [\![\texttt{x}]\!]_{\texttt{a}}$

Program: **Entire Functions**

Preserving the additional structure

$[\![\texttt{Twice}]\!] : \texttt{x} \mapsto \sum_{\texttt{m}} [\![\texttt{Twice}]\!]_{(\texttt{m},\texttt{b})} [\![\texttt{x}]\!]^{\texttt{m}}$

Interaction: **Functional Composition**

Quantitative Semantics: what is it use for ?

Full Abstraction:

Probabilistic Coherent Spaces
&
Probabilistic PCF

with T. Ehrhard and M. Pagani

Full Abstraction

Denotational semantics:
a program as a function between mathematical spaces

Operational semantics:
a program as a sequence of computation steps

$$[\![P]\!] = [\![Q]\!] \quad \underset{\text{Completeness}}{\overset{\text{Adequacy}}{\substack{\Longrightarrow \\ \Longleftarrow}}} \quad P \simeq_o Q$$

$$(\forall C[\,],\ C[P] \to^* v \iff C[Q] \to^* v)$$

Full Abstraction studies connections between denotational and operational semantics. **LCF Considered as a Programming Language, Plotkin (77)**

Full Abstraction

Denotational semantics: Probabilistic Coherent Spaces
a program as a function between mathematical spaces

Operational semantics:
a program as a sequence of computation steps

$$[\![P]\!] = [\![Q]\!] \quad \overset{\text{Adequacy}}{\underset{\text{Completeness}}{\overset{\Longrightarrow}{\Longleftarrow}}} \quad P \simeq_o Q$$

$$(\forall C[\,],\ C[P] \to^* v \iff C[Q] \to^* v)$$

Full Abstraction studies connections between denotational and operational semantics.

LCF Considered as a Programming Language, Plotkin (77)

Probabilistic Coherent Space: $\mathcal{X} = (\mathbb{R}^+)^{|\mathcal{X}|}, \mathrm{P}(\mathcal{X})$
where $|\mathcal{X}|$ is a countable set
and $\mathrm{P}(\mathcal{X}) \subseteq (\mathbb{R}^+)^{|\mathcal{X}|}$

such that the following holds:

closedness: $\mathrm{P}(\mathcal{X})^{\perp\perp} = \mathrm{P}(\mathcal{X})$,

boundedness: $\forall a \in |\mathcal{X}|, \exists \mu > 0, \forall x \in \mathrm{P}(\mathcal{X}), x_a \leq \mu$,

completeness: $\forall a \in |\mathcal{X}|, \exists \lambda > 0, \lambda e_a \in \mathrm{P}(\mathcal{X})$.

Orthogonality: $x, y \in (\mathbb{R}^+)^{|\mathcal{X}|}$.

$$\boxed{x \perp_{\mathcal{X}} y \iff \sum_{a \in |\mathcal{X}|} x_a y_a \leq 1.}$$

The *orthogonal*: $\mathrm{P}(\mathcal{X})^{\perp} = \{y \in (\mathbb{R}^+)^{|\mathcal{X}|} \mid \forall x \in \mathrm{P}(\mathcal{X}), x \perp_{\mathcal{X}} y\}$.

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

Morphisms: $((\mathbb{R}^+)^{|\mathcal{X}|}, \mathrm{P}(\mathcal{X})) \xrightarrow{f} ((\mathbb{R}^+)^{|\mathcal{Y}|}, \mathrm{P}(\mathcal{Y}))$
are **functions** $f : (\mathbb{R}^+)^{|\mathcal{X}|} \to (\mathbb{R}^+)^{|\mathcal{Y}|}$
preserving probabilistic coherence, $f(\mathrm{P}(\mathcal{X})) \subseteq \mathrm{P}(\mathcal{Y})$.

Linear Morphisms: $f(x) = \sum_{a \in |\mathcal{X}|} M(f)_a \cdot x^m$
given by a **matrix** $M(f) \in (\mathbb{R}^+)^{|\mathcal{X}| \times |\mathcal{Y}|}$

Non-Linear Morphisms: $f(x) = \sum_{m \in \mathcal{M}_{\mathrm{fin}}(|\mathcal{X}|)} M(f)_m \cdot x^m$
given by a **matrix** $M(f) \in (\mathbb{R}^+)^{\mathcal{M}_{\mathrm{fin}}(|\mathcal{X}|) \times |\mathcal{Y}|}$

Once : nat → nat

```
λx if x=0 then Coin else 42
```

$$[\![\mathtt{Once\ x}]\!]_0 = [\![\mathtt{Once\ x}]\!]_1 = \frac{1}{2}[\![\mathtt{x}]\!]_0$$
$$[\![\mathtt{Once\ x}]\!]_{42} = \sum_{\mathtt{a}\geq 1}[\![\mathtt{x}]\!]_{\mathtt{a}}$$

Twice : nat → nat

```
λx if x=0 then (if x=0 then Coin else 42)
              else (if x=0 then 42 else 0)
```

$$[\![\mathtt{Twice\ x}]\!]_0 = \frac{1}{2}[\![\mathtt{x}]\!]_0^2 + \sum_{\mathtt{a,b}\geq 1}[\![\mathtt{x}]\!]_{\mathtt{a}}[\![\mathtt{x}]\!]_{\mathtt{b}}$$

$$[\![\mathtt{Twice\ x}]\!]_1 = \frac{1}{2}[\![\mathtt{x}]\!]_0^2 \qquad [\![\mathtt{Twice\ x}]\!]_{42} = 2\sum_{\mathtt{a}\geq 1}[\![\mathtt{x}]\!]_0[\![\mathtt{x}]\!]_{\mathtt{a}}$$

Full Abstraction

Denotational semantics: Probabilistic Coherent Spaces

a program as a function between mathematical spaces

Operational semantics:

a program as a sequence of computation steps

Let $P, Q : \sigma$

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

Adequacy $\Downarrow$ $\Uparrow$ Completeness

$$P \simeq_o Q$$

$$(\forall C[\,],\ C[P] \to^* v \iff C[Q] \to^* v)$$

Full Abstraction studies connections between denotational and operational semantics. **LCF Considered as a Programming Language, Plotkin (77)**

Full Abstraction

Denotational semantics: Probabilistic Coherent Spaces

a program as a function between mathematical spaces

Operational semantics: Probabilistic PCF

a program as a sequence of computation steps

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

Adequacy $\Downarrow$ $\Uparrow$ Completeness

$$P \simeq_o Q$$

$$(\forall C[\ ],\ C[P] \to^* v \iff C[Q] \to^* v)$$

Full Abstraction studies connections between denotational and operational semantics. LCF Considered as a Programming Language, **Plotkin (77)**

A Typed Probabilistic Functional Programing Language

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

Probabilistic PCF:

$$\mathtt{N}, \mathtt{P}, \mathtt{Q} := \underline{\mathtt{n}} \mid \mathsf{pred}(\mathtt{N}) \mid \mathsf{succ}(\mathtt{N}) \mid \mathtt{x} \mid \lambda \mathtt{x}^\sigma\, \mathtt{P} \mid (\mathtt{P})\mathtt{Q} \mid \mathsf{fix}(\mathtt{M})$$
$$\mid \mathsf{if}\ (N = \underline{0})\ \mathsf{then}\ P\ \mathsf{else}\ Q \mid \mathtt{Coin},$$

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

P reduces to Q in one step with probability r

$$\mathtt{Coin} \xrightarrow{1/2} 0$$
$$\mathtt{Coin} \xrightarrow{1/2} 1$$

$$\mathbf{Proba}(P \xrightarrow{*} v) = \sum_{P \xrightarrow{\mathtt{r}_1} \dots \xrightarrow{\mathtt{r}_n} v} r_1 \cdots r_n$$

A Typed Probabilistic Functional Programing Language

Integers :w

$$\underline{n} : \texttt{nat}$$

$$\text{pred}(\underline{k+1}) \xrightarrow{1} \underline{k}$$

$$\text{succ}(\underline{k}) \xrightarrow{1} \underline{k+1}$$

Functions and Composition :

$$(\underset{\sigma\Rightarrow\tau}{\lambda x^\sigma M})\underset{\sigma}{N} \xrightarrow{1} M\,[\underset{\tau}{N/x}]$$

Fixpoints :

$$\text{fix}(M) \xrightarrow{1} (M)\text{fix}(M)$$

Case Zero :

$$\text{if } (\underline{0} = \underline{0}) \text{ then } P_1 \text{ else } P_2 \xrightarrow{1} P_1$$

$$\text{if } (\underline{k+1} = \underline{0}) \text{ then } P_1 \text{ else } P_2 \xrightarrow{1} P_2$$

$+$ Context Rules

Probabilities :

$$\texttt{Coin} \xrightarrow{1/2} 0$$

$$\texttt{Coin} \xrightarrow{1/2} 1$$

where $M \xrightarrow{\texttt{r}} M'$ means that:
M reduces to M' with probability r

Randomized algorithm:

A Las Vegas example.

Input: A $\underline{0}/\underline{1}$ array of length $n \geq 2$ with at least one cell is $\underline{0}$.

$\underline{0}$	$\underline{1}$	$\underline{0}$

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

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

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

This algorithm succeeds with probability one.

- Success in 1 step is : $\frac{2}{3}$.
- Success in 2 steps is : $\frac{2}{3}\ \frac{1}{3}$.
- Success in n steps is : $\frac{2}{3}\ \frac{1}{3^n}$.

...

Success in any steps is :

$$\sum_{k=1}^{\infty} \frac{2}{3}\ \frac{1}{3^k} = 1.$$

Caml encoding:

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

PCF encoding:

$$\textbf{fix}\ \big(\lambda\, \mathtt{LasVegas}^{(\mathtt{nat}\Rightarrow\mathtt{nat})\Rightarrow\mathtt{nat}}$$
$$\lambda \mathtt{f}^{\mathtt{nat}\Rightarrow\mathtt{nat}}$$
$$\Big(\tfrac{1}{n}\lambda \mathtt{g}^{\mathtt{nat}\Rightarrow\mathtt{nat}} g\ \ \underline{0} + \cdots + \tfrac{1}{n}\lambda \mathtt{g}^{\mathtt{nat}\Rightarrow\mathtt{nat}} g\ \ \underline{n-1}\Big)$$
$$\lambda \mathtt{k}^{\mathtt{nat}}\ \mathtt{if}\ (\mathtt{f}\ \ \mathtt{k} = \underline{0})\ \mathtt{then}\ \mathtt{k}$$
$$\mathtt{else}\ \mathtt{LasVegas}\ \mathtt{f}\ \big)$$

Input: A $\underline{0}/\underline{1}$ array of length $n \geq 2$ with at least one cell is $\underline{0}$.

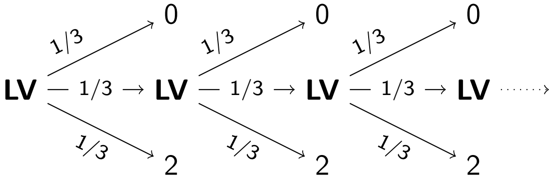
$\underline{0}$	$\underline{1}$	$\underline{0}$

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

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

$$
\begin{aligned}
\textbf{LV} = \textbf{fix}\, \big(\lambda\ & \mathtt{LasVegas}^{(\mathtt{nat}\Rightarrow\mathtt{nat})\Rightarrow\mathtt{nat}} \\
& \lambda \mathtt{f}^{\mathtt{nat}\Rightarrow\mathtt{nat}}\ \big(\tfrac{1}{3}\lambda \mathtt{g}\ \mathtt{g}\ \underline{0} + \tfrac{1}{3}\lambda \mathtt{g}\ \mathtt{g}\ \underline{1} + \tfrac{1}{3}\lambda \mathtt{g}\ \mathtt{g}\ \underline{2}\big) \\
& \qquad \lambda \mathtt{k}^{\mathtt{nat}}\ \mathtt{if}\ (\mathtt{f}\ \mathtt{k} = \underline{0})\ \mathtt{then}\ \mathtt{k} \\
& \qquad\quad \mathtt{else}\ \ \mathtt{LasVegas}\ \mathtt{f}\big)
\end{aligned}
$$

Operational Semantics:



Full Abstraction

Denotational semantics: Probabilistic Coherent Spaces
a program as a function between mathematical spaces

Operational semantics: Probabilistic PCF
a program as a sequence of computation steps

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

Adequacy $\Downarrow$ $\Uparrow$ Completeness

$$\forall C : \sigma \Rightarrow \texttt{nat},\ \forall n \in |\texttt{nat}|,$$
$$\mathrm{Proba}((C)P \xrightarrow{*} n) = \mathrm{Proba}((C)Q \xrightarrow{*} n))$$

Full Abstraction studies connections between denotational and operational semantics. **LCF Considered as a Programming Language, Plotkin (77)**

Full Abstraction

Denotational semantics: Probabilistic Coherent Spaces
a program as a function between mathematical spaces

Operational semantics: Probabilistic PCF
a program as a sequence of computation steps

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

Adequacy $\Downarrow$ $\Uparrow$ Completeness

$$\forall C : \sigma \Rightarrow \texttt{nat},\ \forall n \in |\texttt{nat}|,$$
$$\mathrm{Proba}((C)P \xrightarrow{*} n) = \mathrm{Proba}((C)Q \xrightarrow{*} n))$$

Full Abstraction studies connections between denotational and operational semantics. **LCF Considered as a Programming Language, Plotkin (77)**

Let $P, Q : \texttt{nat}$	$\forall n \in \mathbb{N},\ [\![P]\!]_n = [\![Q]\!]_n$
	Adequacy $\Downarrow \Uparrow$ Completeness
$\forall C : \texttt{nat} \Rightarrow \texttt{nat},\ \forall n \in \mathbb{N},$	$\mathrm{Proba}((C)P \xrightarrow{*} n) = \mathrm{Proba}((C)Q \xrightarrow{*} n))$

Adequacy Lemma:[DanosEhrhard] $\forall n,\ \mathrm{Proba}(P \xrightarrow{*} \underline{n}) = [\![P]\!]_n.$

By contradiction:

$$[\![P]\!]_n \neq [\![Q]\!]_n \Rightarrow \mathrm{Proba}(P \xrightarrow{*} \underline{n}) \neq \mathrm{Proba}(Q \xrightarrow{*} \underline{n})$$

Assumption: Let $P, Q : \phi \Rightarrow \psi$,
$\exists \alpha = ([\gamma_1, \ldots, \gamma_n], \beta)$ such that $[\![P]\!]_\alpha \neq [\![Q]\!]_\alpha$.

Goal: Find $T_\alpha : (\phi \Rightarrow \psi) \to nat$ s.t.
$\mathrm{Proba}((T_\alpha)P \xrightarrow{*} \underline{0}) \neq \mathrm{Proba}((T_\alpha)Q \xrightarrow{*} \underline{0})$

Assumption: Let $P, Q : \phi \Rightarrow \psi$,

$\exists\alpha = ([\gamma_1, \ldots, \gamma_n], \beta)$ such that $[\![P]\!]_\alpha \neq [\![Q]\!]_\alpha$.

Goal: Find $T_\alpha : (\phi \Rightarrow \psi) \to \mathit{nat}$ s.t. $[\![(T_\alpha)P]\!]_0 \neq [\![(T_\alpha)Q]\!]_0$

$\mathrm{Proba}((T_\alpha)P \xrightarrow{*} \underline{0}) \neq \mathrm{Proba}((T_\alpha)Q \xrightarrow{*} \underline{0})$

Assumption: Let $P, Q : \phi \Rightarrow \psi$,

$\exists \alpha = ([\gamma_1, \ldots, \gamma_n], \beta)$ such that $[\![P]\!]_\alpha \neq [\![Q]\!]_\alpha$.

Choose Testing Context and add **Parameters**

$$\mathcal{T}_\alpha(\vec{r}) = \lambda f^{\phi \Rightarrow \psi}\, \mathcal{T}_\beta(\vec{r}')\left((f)\sum_{i=1}^{n} \frac{r_i}{n} \mathcal{N}_{\gamma_i}(\vec{r}'_i)\right)$$

$$\mathcal{N}_\alpha(\vec{r}) = \lambda x^\phi \,\text{if}\, (\wedge_{i=1}^{k} \mathcal{T}_{\gamma_i}(\vec{r}_i) x = \underline{0})\,\text{then}\, \mathcal{N}_\beta(\vec{r}')\,\text{else}\, \Omega_\psi.$$

Goal: Find $T_\alpha : (\phi \Rightarrow \psi) \to nat$ s.t. $[\![(T_\alpha)P]\!]_0 \neq [\![(T_\alpha)Q]\!]_0$

$\text{Proba}((T_\alpha)P \xrightarrow{*} \underline{0}) \neq \text{Proba}((T_\alpha)Q \xrightarrow{*} \underline{0})$

Assumption: Let $P, Q : \phi \Rightarrow \psi$,

$\exists \alpha = ([\gamma_1, \ldots, \gamma_n], \beta)$ such that $[\![P]\!]_\alpha \neq [\![Q]\!]_\alpha$.

Choose Testing Context and add **Parameters**

$$\mathcal{T}_\alpha(\vec{r}) = \lambda f^{\phi \Rightarrow \psi}\, \mathcal{T}_\beta(\vec{r}') \left((f) \sum_{i=1}^{n} \frac{r_i}{n} \mathcal{N}_{\gamma_i}(\vec{r}_i') \right)$$

$$\mathcal{N}_\alpha(\vec{r}) = \lambda x^\phi \,\mathrm{if}\, (\wedge_{i=1}^{k} \mathcal{T}_{\gamma_i}(\vec{r}_i) x = \underline{0})\, \mathrm{then}\, \mathcal{N}_\beta(\vec{r}')\, \mathrm{else}\, \Omega_\psi.$$

Observe by induction:

- $[\![(\mathcal{T}_\alpha(\vec{r}))M]\!]_0$ is entire with finitely many parameters (d_α).
- If $0 < \vec{r} < 1$ are dyadic reals, then $\mathcal{T}_\alpha(\vec{r})$ is in PPCF.
- The coefficient of $\prod \vec{r}$ is proportional to $[\![M]\!]_\alpha$.

Goal: Find $T_\alpha : (\phi \Rightarrow \psi) \rightarrow nat$ s.t. $[\![(T_\alpha)P]\!]_0 \neq [\![(T_\alpha)Q]\!]_0$

$\mathrm{Proba}((T_\alpha)P \xrightarrow{*} \underline{0}) \neq \mathrm{Proba}((T_\alpha)Q \xrightarrow{*} \underline{0})$

Assumption: Let $P, Q : \phi \Rightarrow \psi$,

$\exists \alpha = ([\gamma_1, \ldots, \gamma_n], \beta)$ such that $[\![P]\!]_\alpha \neq [\![Q]\!]_\alpha$.

Choose Testing Context and add **Parameters**

$$\mathcal{T}_\alpha(\vec{r}) = \lambda f^{\phi \Rightarrow \psi}\, \mathcal{T}_\beta(\vec{r}') \left((f) \sum_{i=1}^{n} \frac{r_i}{n} \mathcal{N}_{\gamma_i}(\vec{r}_i') \right)$$

$$\mathcal{N}_\alpha(\vec{r}) = \lambda x^\phi \,\text{if}\, (\wedge_{i=1}^{k} \mathcal{T}_{\gamma_i}(\vec{r}_i) x = \underline{0}) \,\text{then}\, \mathcal{N}_\beta(\vec{r}') \,\text{else}\, \Omega_\psi .$$

Observe by induction:

- $[\![(\mathcal{T}_\alpha(\vec{r}))M]\!]_0$ is entire with finitely many parameters (d_α).
- If $0 < \vec{r} < 1$ are dyadic reals, then $\mathcal{T}_\alpha(\vec{r})$ is in PPCF.
- The coefficient of $\prod \vec{r}$ is proportional to $[\![M]\!]_\alpha$.

Entire series: $[\![(\mathcal{T}_\alpha(\vec{r}))P]\!]_0$ and $[\![(\mathcal{T}_\alpha(\vec{r}))Q]\!]_0$ are entire $\mathbb{R}^{d_\alpha} \to \mathbb{R}$ with different coefficients.

Goal: Find $T_\alpha : (\phi \Rightarrow \psi) \to nat$ s.t. $\quad [\![(T_\alpha)P]\!]_0 \neq [\![(T_\alpha)Q]\!]_0$

$\mathrm{Proba}((T_\alpha)P \xrightarrow{*} \underline{0}) \neq \mathrm{Proba}((T_\alpha)Q \xrightarrow{*} \underline{0})$

Related Works:

Weighted Relational Models of Typed λ-calculi [LairdManzonettoMcCuskerPagani]

$\mathbb{R}^+ \cup \infty$
Not well pointed.
Fully Abstract for probabilistic PCF

Probabilistic Games [DanosHarmer]

Keep order of inputs.
Definability result followed by the extentional collapse.
Fully Abstract for probabilistic idealized algol (with references)

Probabilistic monads [PlotkinJones]

A model of first-order call by value language

Quantitative Semantics: what is it use for ?

A convenient model of

Functional computation

&

Derivation

with M. Kerjean

Interaction Syntax and Semantics:

- Stable functions & Linear Logic $A \Rightarrow B = !A \multimap B$
- Quantitative Semantics & Differential lambda-calculus
- Differential equations & ??

A convenient category for mathematics...

that is a category of topological spaces which is

Cartesian Closed;

Complete and Cocomplete.

... and for classical linear logic $(\neg\neg A = A)$.

The Convenient Global Setting for analysis

Most results are extracted or adapted from [Michor and Kriegl]

Let E, F be locally convex topological vector spaces (tvs).

Bounded sets: $B \subseteq E$ absorbed by any open, up to dilatation.

$$B \text{ bounded} \iff \forall V,\ \exists \rho \text{ s.t. } B \subseteq \rho V$$

Bounded maps: $f : E \to F$ preserving bounded sets.

$$\forall B \text{ bounded in } E,\ f(B) \text{ is bounded in } F$$

Bounded equivalence: $E \simeq F$ iff there is a bijection $E \xrightarrow{\phi} F$ such that ϕ and ϕ^{-1} are bounded linear.

Linear Category of Reflexive Spaces: **Definition**

Bounded dual: $E^\times$ is the lcts of *bounded linear* forms, endowed with the *bounded open* topology:

$$E^\times = \{\phi : E \to \mathbb{C} \mid \phi \text{ bounded linear}\}$$
$$\forall B, \epsilon,\ \mathcal{W}(B,\epsilon) = \{\phi : E \to \mathbb{C} \mid \forall x \in B,\ |\phi(x)| \leq \epsilon\}$$

Linear Category of Reflexive spaces:

Objects: E lcts s.t. $E^{\times\times} \simeq E$

Maps: $\mathrm{Lin}(E,F) = \{\phi : E \to F \text{ bounded linear}\}$

Examples: $[\![\texttt{bool}]\!] = \mathbb{C} \oplus \mathbb{C}$ and $[\![\texttt{nat}]\!] = \mathbb{C}^{(\mathbb{N})}$

Counter-Examples: c_0, l^1 and l^∞.

BiProduct $E \times F$ and $B_E \times B_F$

Diagonal $\left\{ \begin{array}{ccc} E & \to & E \times E \\ x & \mapsto & (x, x) \end{array} \right.$ Codiagonal $\left\{ \begin{array}{ccc} E \times E & \to & E \\ (x, y) & \mapsto & x + y \end{array} \right.$

Accessible Products and Coproducts

$\bigoplus_{i \in I} E_i$ and finite sum of bounded

$\prod_{i \in I} E_i$ and infinite product of bounded

Linear Function Space $\mathrm{Lin}(E, F)$ and equibounded $\mathcal{B}$

$\forall B_E, \exists B_F, \forall f \in \mathcal{B},\ f(B_E) \subseteq B_F$

Tensor Product $E \otimes F$ as vector spaces and $B_E \otimes B_F$

$$\begin{array}{ccc} (E \otimes F)^{\times} & \simeq & \mathrm{Lin}(E, F^{\times}) \\ \phi & \mapsto & \lambda x\, [\lambda y\, \phi(x \otimes y)] \end{array}$$

Theorem: **Lin** is Symetric Monoidal Closed.

Proof Sketch

- Bounded version of Hahn Banach
- Reflexive spaces are bounded complete:
  E_B, the span of any bounded B, is a Banach Space.
- Bounded Banach Steinhaus (equibounded = simply bounded).
- $(\mathrm{Lin}(E, F))^\times \overset{\sim}{\hookrightarrow} (\prod_{a \in E} F)^\times \simeq \oplus_{a \in E} F$
- Tensor and Linear function spaces preserve reflexivity.

NonLinear Category of Reflexive Spaces:

n-Monomial: $f_n : E \to F$ n-homogene $f_n(tx) = t^n f_n(x)$.

Ser(E,F) the reflexive space of bounded entire functions $f = \sum_n f_n$
uniformly converging on bounded sets.

Bounded Open Topology

$$\mathcal{W}(B_E, V_F) = \{f \in \mathrm{Ser}(E, F) \mid f(B_E) \subseteq V_F\}$$

Equibounded Bornology

$\mathcal{B}$ s.t. $\forall B_E, \exists B_F, f \in \mathcal{B} \Rightarrow f(B_E) \subseteq B_F$

Non Linear Category

Objects: E lcts s.t. $E^{\times\times} \simeq E$

Maps: $\mathrm{Ser}(E, F)$

Theorem: **Ser** is Cartesian Closed.

Exponential Functor:

$$!E = \mathrm{Ser}(E, \mathbb{C})^{\times} \quad | \quad \begin{array}{rrcl} !f: & !E & \to & !F = \mathrm{Ser}(F, \mathbb{C})^{\times} \\ & \phi & \mapsto & (F \xrightarrow{h} \mathbb{C}) \mapsto \langle \phi \ , \ E \xrightarrow{f} F \xrightarrow{h} \mathbb{C} \rangle \end{array}$$

Dirac Mass:

$$\forall x \in E, \ \delta_x : f \mapsto f(x) \in !E \quad | \quad !E = \overline{\mathrm{span}(\delta_x \ | x \in E)}^{\mathrm{B}}$$

A comonad:

$$\begin{array}{rrcl} \delta: & E & \to & !E \\ & x & \mapsto & \delta_x \end{array} \quad | \quad \begin{array}{rrcl} \mu: & !!E & \to & !E = \mathrm{Ser}(E, \mathbb{C})^{\times} \\ & \phi & \to & (E \xrightarrow{f} \mathbb{C}) \mapsto \langle \phi \ , \ \delta_f \rangle \end{array}$$

Free comonoid:

$$\begin{array}{rrcl} c: & !E \otimes !E & \to & !E \\ & \delta_x \otimes \delta_y & \to & \delta_{x+y} \end{array} \quad | \quad \begin{array}{rrcl} w: & !E & \to & 1 = \mathbb{C} \\ & \phi & \to & \langle \phi \ , \ \lambda x 1 \rangle \end{array}$$

LL Theorems:

$$\mathrm{Ser}(E, F) \simeq \mathrm{Lin}(!E, F) \quad | \quad !E \otimes !F \simeq !(E \times F)$$

$$\begin{array}{ccc} E & \longmapsto & !E \\ f : E \Rightarrow F & & f^! : !E \multimap !F,\ f^!(\delta_x) = f(x) \end{array}$$

$$\begin{tikzcd} \textbf{ccc} & \textbf{Ser} \arrow[rr, bend left, "\mathrm{L}"] \arrow[rr, phantom, "\perp"] & & \textbf{Lin} \arrow[ll, bend left, "\mathrm{Forget}"] & \textbf{smcc} \end{tikzcd}$$

$$\begin{array}{ccc} E & \longmapsfrom & E \\ f : E \Rightarrow F & & f : E \multimap F \end{array}$$

Some inhabitants of !E:

$$\mathrm{Ser}(E,\mathbb{C})^{\times}$$

$$\forall x \in E,\ \delta_x : f \mapsto f(x)$$

$$\theta_n(x) : \sum_n f_n \mapsto f_n(x)$$

Taylor expansion:

$$\delta_x = \sum \theta_n(x) \qquad \mathrm{Ser}(E,F) = \overline{\mathrm{Pol}_n(E,F)}^B = \overline{\oplus_n\ \tilde{\otimes}^n E}^B$$

(equibounded)

Bialgebra structure:

$$\begin{array}{ccc} !E & \to & !E \otimes !E \\ \phi & \mapsto & \phi \otimes \phi \end{array} \qquad \begin{array}{ccc} !E \otimes !E & \xrightarrow{\overline{c}_E} & !E \\ h & \mapsto & \phi(\lambda x\ \psi(\lambda y\ h(x+y))) \end{array}$$

A derivation operator:

$$\overline{d}_E \in \mathrm{Lin}(E, !E)$$

$$\begin{array}{ccccccc} !E \otimes E & \xrightarrow{-\otimes \overline{d}_E} & !E \otimes !E & \xrightarrow{\overline{c}_E} & !E & \xrightarrow{f} & F \\ \delta_x \otimes y & \mapsto & \delta_x \otimes \lim\limits_{t\to 0} \frac{\delta_{ty} - \delta_0}{t} & \mapsto & \lim\limits_{t\to 0} \frac{\delta_{x+ty} - \delta_x}{t} & \mapsto & \lim\limits_{t\to 0} \frac{f(x+ty) - f(x)}{t} \end{array}$$

Related Works:

Fock Spaces [BlutePanangadenSeely]

Banach Spaces and contractive maps

A model of weakening

Köthe Spaces and Finiteness Spaces [Ehrhard]

Sequence spaces, continuous linear and entire functions

Convenient Vector Spaces [BluteEhrhardTasson]

CVS, bounded continuous linear maps, and smooth functions

Applying Quantitative Semantics to Higher-Order Quantum Computing [PaganiValiron]

Coefficients are Positive Matrices

What is it ?

- Linear Logic approach to semantics
- Topological vector spaces, Linear maps and Entire functions

What is it used for ?

- *PCoh Spaces are fully abstract for Probabilist PCF*, with T. Ehrhard and M. Pagani
- *A convenient model of lambda calculus*, Master dissertation of M. Kerjean

Postdoc hiring with M. Pagani

- COmputing with QUAntitative Semantics [CoQuaS]
- `http://lipn.univ-paris13.fr/~pagani/pmwiki/pmwiki.php/Coquas/Coquas`