Linearity, from Mathematics to Computer Science

Christine Tasson
tasson@pps.jussieu.fr

Laboratoire Preuves Programmes Systèmes
Université Paris Diderot
France

Kurims's Computer Science Seminar
25 July 2008

Introduction

1987 : Girard introduces Linear Logic.

1988 : Girard links denotational semantics to power series.

2001 : Ehrhard and Regnier introduce differential lambda-calculus.

2005 : Ehrhard and Regnier present differential nets.

Summary

The Question

How many times a program uses its argument ?

Let's look at an example :

Power :

$$\left\{ \begin{array}{ccccc} \mathbb{R} & \times & \mathbb{N} & \to & \mathbb{R} \\ x & , & n & \mapsto & x^n \end{array} \right.$$

```
let rec power x n =
  match n with
    | 0 -> 1
    | n -> x * (power x (n-1))
```

Power uses its first argument several times and its second one only once.

Semantics

Model

A program is interpreted using mathematical objects.

$$[Prog] : A \Rightarrow B$$

Linear Logic

Every program can be decomposed into an exponential part (! which means the ressource is infinite) and a linear part ($\multimap$ which means the program consults its ressource only once).

$$[Prog] : !A \multimap B$$

For instance, Power :

$$\begin{array}{ccccc} !\mathbb{R} & \otimes & \mathbb{N} & \multimap & \mathbb{R} \\ (x & , & n) & \mapsto & x^n \end{array}$$

An Analogy

Mathematical Linearity

A linear function is a first degree polynomial function.

Every regular function can be approximated by a linear function :

$$f(x) \underset{x \to 0}{\simeq} f(0) + f'(0)\,x$$

Computer Science Linearity

A linear program is a program which uses its argument at most once, that is a lambda term $\lambda x \cdot t$ where the variable x appears only once in x.

$$D(\lambda x \cdot t)(s) = t[x \backslash s]_{\text{linear}}$$

Differential analysis

Taylor expansion

An analytic function can be decomposed into a sum of degree n polynomial functions :

$$f(x) = \sum_n \frac{f^{(n)}(0)}{n!} x^n$$

Computer Science version

How can we decompose a program into n-linear ones (which respectively uses its argument exactly n times) ?

Summary

An extension of λ-Calculus

Syntax

$$\begin{aligned} s, t &:= x \mid \lambda x.s \mid (s)t \mid Ds.t \mid 0 \mid as + bt \\ a, b &\in R \text{ where } R \text{ is a ring.} \end{aligned}$$

New ingredients

- 0 means a *deadlock* has been reached.
- *Differentiation operator* $Ds.t$ means the linear application of s to t.
- *Sums* similar to non determinism.

Linear Analogy and Sums

$$\begin{aligned} \lambda x.(s+t) &= \lambda x.s + \lambda x.t & (1) \\ (s+t)u &= (s)u + (t)u & (2) \\ (s)(u+v) &\neq (s)u + (s)v & (3) \end{aligned}$$

Mathematics linearity

Linearity means commutation with sums. The point (3) has to be related with analytic functions semantics.

Linear Analogy and Sums

$$\lambda x.(s+t) \rightarrow \lambda x.s + \lambda x.t \quad (1)$$
$$(s+t)u \rightarrow (s)u + (t)u \quad (2)$$
$$(s)(u+v) \not\rightarrow (s)u + (s)v \quad (3)$$

Non-deterministic quasi-reduction

Intuitively, $s + s'$ reduces on both s and s'. The point (3) comes from s can need its argument several times.
For instance :

$$(\lambda x.(x)x)(\lambda x.x + \lambda x.y) \rightarrow \lambda x.x + \lambda x.y + 2y$$

Notice that y appears two times in the result.

Substitutions and Differentiation

Differential reduction

$$D(\lambda x.t).u \to \lambda x.\left(\frac{\partial t}{\partial x}.u\right) \tag{4}$$

Linear substitution :

The term $\frac{\partial t}{\partial x}.u$ means one occurence of x has been substituted by u in t. It is a non deterministic operation since there are several occurencies that can be substituted.

Substitutions and Differentiation

Differential reduction

$$D(\lambda x.t).u \rightarrow \lambda x.\left(\frac{\partial t}{\partial x}.u\right) \qquad (4)$$

Linear substitution :

The term $\frac{\partial t}{\partial x}.u$ means one occurence of x has been substituted by u in t. It is a non deterministic operation since there are several occurencies that can be substituted.

$$\frac{\partial y}{\partial x}.u \quad = \quad \delta_{xy} u$$

Substitutions and Differentiation

Differential reduction

$$D(\lambda x.t).u \rightarrow \lambda x.\left(\frac{\partial t}{\partial x}.u\right) \tag{4}$$

Linear substitution :

The term $\frac{\partial t}{\partial x}.u$ means one occurence of x has been substituted by u in t. It is a non deterministic operation since there are several occurencies that can be substituted.

$$\frac{\partial (s) t}{\partial x}.u \quad = \quad \left(\frac{\partial s}{\partial x}.u\right) t + Ds.\left(\frac{\partial t}{\partial x}.u\right)$$

Substitutions and Differentiation

Differential reduction

$$D(\lambda x.t).u \to \lambda x.\left(\frac{\partial t}{\partial x}.u\right) \quad (4)$$

Linear substitution :

The term $\frac{\partial t}{\partial x}.u$ means one occurence of x has been substituted by u in t. It is a non deterministic operation since there are several occurencies that can be substituted.

$$\frac{\partial (s) t}{\partial x}.u \;=\; \left(\frac{\partial s}{\partial x}.u\right) t + Ds.\left(\frac{\partial t}{\partial x}.u\right)$$

$$\to (f \circ g)'(x) = f'(g(x)) \cdot g'(x)$$

Substitutions and Differentiation

Differential reduction

$$D(\lambda x.t).u \to \lambda x.\left(\frac{\partial t}{\partial x}.u\right) \tag{4}$$

Linear substitution :

The term $\frac{\partial t}{\partial x}.u$ means one occurence of x has been substituted by u in t. It is a non deterministic operation since there are several occurencies that can be substituted.

$$\frac{\partial s[x_1, x_2 \leftarrow x]}{\partial x}.u \quad = \quad \left(\frac{\partial s}{\partial x_1}.u\right)[x_1, x_2 \leftarrow x] + \left(\frac{\partial s}{\partial x_2}.u\right)[x_1, x_2 \leftarrow x])$$

Substitutions and Differentiation

Differential reduction

$$D(\lambda x.t).u \rightarrow \lambda x.\left(\frac{\partial t}{\partial x}.u\right) \tag{4}$$

Linear substitution :

The term $\frac{\partial t}{\partial x}.u$ means one occurence of x has been substituted by u in t. It is a non deterministic operation since there are several occurencies that can be substituted.

$$\frac{\partial s[x_1, x_2 \leftarrow x]}{\partial x}.u \;=\; \left(\frac{\partial s}{\partial x_1}.u\right)[x_1, x_2 \leftarrow x] + \left(\frac{\partial s}{\partial x_2}.u\right)[x_1, x_2 \leftarrow x])$$

$$\rightarrow (f.g)' = f'.g + f.g'$$

Reduction

Definition

The smallest reduction closed by context and by sums that contains both :

β-reduction	$(\lambda x.s)u$	$\rightarrow$	$s[x/u]$
Differential reduction	$D(\lambda x.t).u$	$\rightarrow$	$\lambda x.(\frac{\partial t}{\partial x}.u)$

Theorem (Ehrhard, Régnier 2001)

This reduction is confluent and if the ring is $\mathbb{N}$, simply typed terms are strongly normalizing.

Taylor expansion

Definition

Usual application can be encoded using differential application :

$$(s)u = \sum_{n=0}^{\infty} \frac{1}{n!}(D^n s.u^n)0 \qquad (5)$$

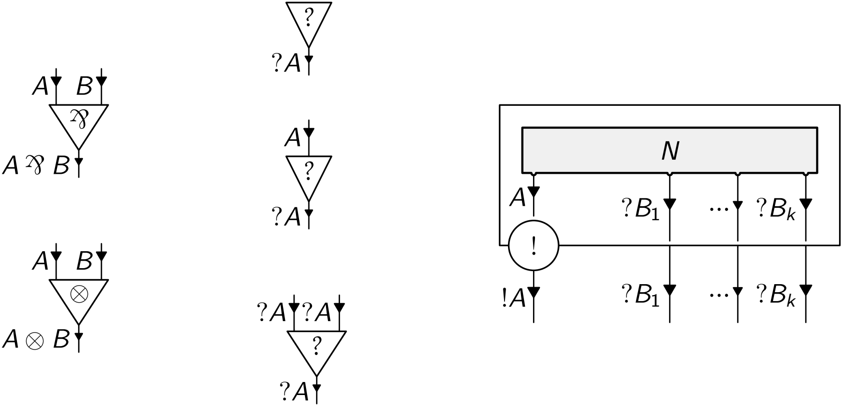
Theorem (Ehrhard, Régnier 2006)

Purely λ-calculus can be encoded through Taylor Expansion in the purely differential λ-calculus.

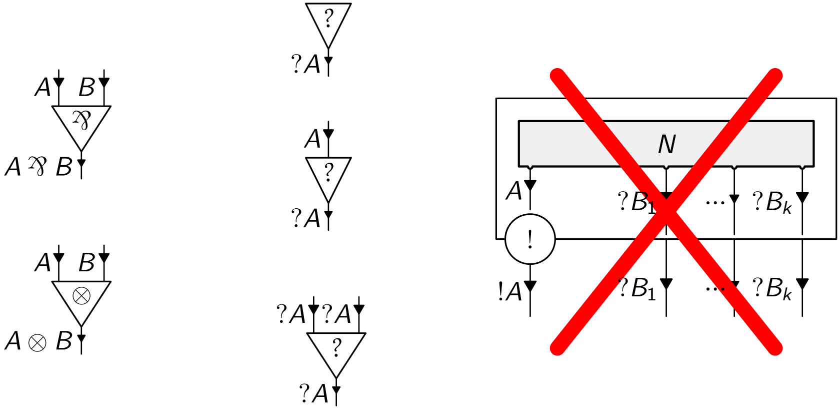
Summary

Linear Logic Nets

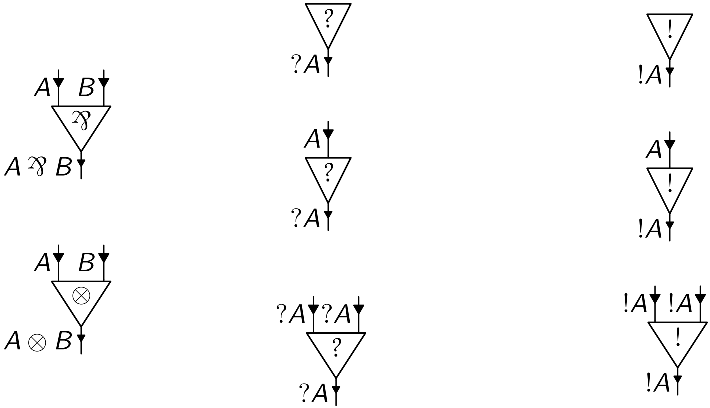
A programming language :



A Linearized programming language :



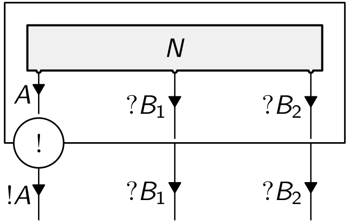
A Linearized programming language :



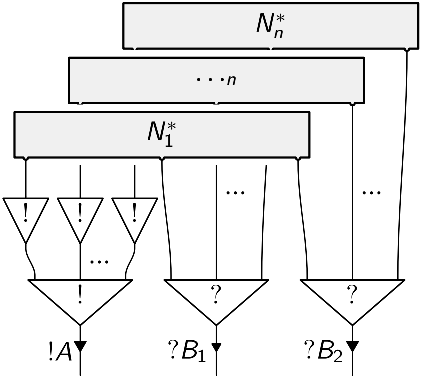
Taylor and Computer Science

The principle :

To every linear net N and for every n, corresponds a differential net that appears in the taylor expansion.



$$\rightsquigarrow \sum_{n=0}^{\infty} \frac{1}{n!}$$



where N_k^* in Taylor expansion of N.

Differential Nets vs. Differential λ-Calculus

Theorem (Ehrhard, Régnier 2006)

Differential λ-calculus can be encoded in Differential nets in such a manner that the first reduction is simulated by the second.

Advantages of Differential nets

- An extension conservative of differential λ-calculus.
- Symmetry between ?- and !-cells that is the monad and the comonad.
- Links with concurrence : π-calculus can be encoded in differential nets.

Summary

History of linear models

Linear Logic

	A	$\lvert A\rvert$	$[A] = \mathbf{k}^{\lvert A\rvert}$
$\perp$	$A^{\perp}$	$\lvert A\rvert$	$\mathcal{L}([A], \mathbf{k})$
$\oplus, \&$	$A \oplus B$	$\lvert A\rvert + \lvert B\rvert$	$[A] \oplus [B]$
$\otimes$	$A \otimes B$	$\lvert A\rvert \times \lvert B\rvert$	$[A] \otimes [B]$
$\multimap$	$A \multimap B$	$\lvert A\rvert \times \lvert B\rvert$	$\mathcal{L}([A], [B])$
$!$	$!A$	$\mathcal{M}_f(\lvert A\rvert)$	??

Models

- The simplest is the model of sets and relations.
- Taking sets as bases and relations as matrices support, we get the model of linear spaces.
- Because of exponential, infinite dimension is needed.

Bibliography

Infinite dimension problems

- Which basis notion ?
- How to ensure reflexivity ?

In order to solve them, we need some topology.

[Blute] *Linear Lauchli semantics*, Annals of Pure and Applied Logic, 1996

[Girard] *Coherent Banach spaces*, Theoretical Computer Science, 1999

[Ehrhard] *On Köthe sequence spaces and linear logic*, Mathematical Structures in Computer Science, 2002

[Ehrhard] *Finiteness spaces*, Mathematical Structures in Computer Science, 2005

The relational model view point.

Definition

Let $|X|$ be countable, for each $\mathcal{F} \subseteq \mathcal{P}(|X|)$, let us denote

$$\mathcal{F}^{\perp} = \{u' \subseteq |X| \,|\, \forall u \in \mathcal{F},\ u \cap u' \text{ finite}\}.$$

A *finiteness space* is a pair $X = (|X|, \mathcal{F}(X))$ such that $\mathcal{F}(X)^{\perp\perp} = \mathcal{F}(X)$.

Example : Integers.

The linear spaces view point.
For every $x \in \mathbf{k}^{|X|}$, the *support* of x is $|x| = \{a \in |X| \, | \, x_a \neq 0\}$.

Definition

The *linear space* associated to $X = (|X|, \mathcal{F}(X))$ is :

$$\mathbf{k}\langle X \rangle = \{x \in \mathbf{k}^{|X|} \, | \, |x| \in \mathcal{F}(X)\}.$$

endowed by the *topology* generated by the basis at zero :
$\{V_J \, | \, J \in \mathcal{F}^{\perp}\}$ where

$$V_J = \{x \in \mathbf{k}\langle X \rangle \, | \, |x| \cap J = \emptyset\}.$$

Example : Integers.

A Linear Logic Model

$$X^\perp \rightsquigarrow \mathbf{k}\langle X\rangle'$$

$$0 \rightsquigarrow \{0\}$$

$$\left.\begin{array}{c} X \& Y \\ X \oplus Y \end{array}\right\} \rightsquigarrow \mathbf{k}\langle X\rangle \oplus \mathbf{k}\langle Y\rangle$$

$$1 \rightsquigarrow \mathbf{k}$$

$$X \multimap Y \rightsquigarrow \mathcal{L}_c(X, Y)$$

$$X \otimes Y \rightsquigarrow \mathbf{k}\langle X\rangle \otimes \mathbf{k}\langle Y\rangle$$

$$!X \rightsquigarrow \mathbf{k}\langle !X\rangle$$

$$|!X| = \mathcal{M}_{fin}(|X|)$$

where $\mathcal{F}(!X) = \{A \subseteq \mathcal{M}_{fin}(|X|) \mid \bigcup_{m \in A} |m| \in \mathcal{F}X\}$

A Linear Logic Model

$$X^\perp \quad \rightsquigarrow \quad \mathbf{k}\langle X\rangle' \qquad \Rightarrow \text{Reflexivity}$$

$$0 \quad \rightsquigarrow \quad \{0\}$$

$$\left.\begin{array}{c} X \& Y \\ X \oplus Y \end{array}\right\} \quad \rightsquigarrow \quad \mathbf{k}\langle X\rangle \oplus \mathbf{k}\langle Y\rangle$$

$$1 \quad \rightsquigarrow \quad \mathbf{k}$$

$$X \multimap Y \quad \rightsquigarrow \quad \mathcal{L}_c(X, Y)$$

$$X \otimes Y \quad \rightsquigarrow \quad \mathbf{k}\langle X\rangle \otimes \mathbf{k}\langle Y\rangle$$

$$!X \quad \rightsquigarrow \quad \mathbf{k}\langle !X\rangle \qquad \Rightarrow \text{Infinite dimension}$$

$$|!X| = \mathcal{M}_{fin}(|X|)$$

where $\mathcal{F}(!X) = \{A \subseteq \mathcal{M}_{fin}(|X|) \mid \bigcup_{m \in A} |m| \in \mathcal{F}X\}$

Finiteness Spaces

Theorem

Finiteness spaces are a model of differential nets.

Taylor expansion

A program of type : $A \Rightarrow B$ is interpreted by an analytic function.

Finiteness Spaces

Theorem

Finiteness spaces are a model of differential nets.

Differential nets have been designed to correspond to this semantics.

Taylor expansion

A program of type : $A \Rightarrow B$ is interpreted by an analytic function.

This analytic function embodies the analogy between mathematics linearity and computer science linearity.

Lefschetz and al

Linearized topological vector spaces have been introduced by S. Lefschetz in 1942.

They appear in

- [Barr] *⋆-autonomous Categories*, Lecture Notes in Mathematics, 1979
- [Blute] *Linear Lauchli semantics*, Annals of Pure and Applied Logici, 1996
- [Ehrhard] *Finiteness spaces*, Mathematical Structures in Computer Science, 2005

Lefschetz spaces

Definition

Let $E, \mathcal{T}$ be topological $\mathbf{k}$-vector space.
E is said to be a *Lefschetz space* if :

- $\mathbf{k}$ is discrete.
- There is a filter basis at zero $\mathcal{V}$ which generates $\mathcal{T}$ and which is made of linear subspaces.
- $\bigcap \mathcal{V} = \{0\} \quad \Rightarrow \quad$ Hausdorff topology.

Example : Finiteness spaces with the basis topology.
Finite sequences $\mathbf{k}^{(\omega)}$ with finite codimension topology.

Lefschetz spaces

Definition

Let $E, \mathcal{T}$ be topological $\mathbf{k}$-vector space.
E is said to be a *Lefschetz space* if :

- $\mathbf{k}$ is discrete.
- There is a filter basis at zero $\mathcal{V}$ which generates $\mathcal{T}$ and which is made of linear subspaces.
- $\bigcap \mathcal{V} = \{0\} \quad \Rightarrow \quad$ Hausdorff topology.

Example : Finiteness spaces with the basis topology.
Finite sequences $\mathbf{k}^{(\omega)}$ with finite codimension topology.

This topology is counter intuitive

- A finite dimension Lefschetz space is discrete.
- Open bowls are affine subspaces.
- Open linear subspaces are closed.

Function spaces and Orthogonal

Definition (Linear compactness)

A subspace K of a Lefschetz Space is said *linearly compact* when for every closed affine filter $\mathcal{F} = \{F_\alpha\}$ satisfying the intersection property ($\forall F_\alpha,\ F_\alpha \cap K \neq \emptyset$),

$$(\cap\mathcal{F}) \cap K \neq \emptyset.$$

Definition (Compact open topology)

This is the topology of uniform convergence on linearly compact subspaces.

Bases at zero

- Functionals $\mathcal{L}_c(E, F)$: $W(K, V) = \{f \,|\, f(K) \subset V\}$ with K linear compact and V open subspace.
- Dual space E' : $K^\perp = \{x' \,|\, \forall x \in K,\ x'(x) = 0\}$ with K linearly compact subspace.

Definition (Linear compactness)

A subspace K of a Lefschetz Space is said *linearly compact* when for every closed affine filter $\mathcal{F} = \{F_\alpha\}$ satisfying the intersection property ($\forall F_\alpha$, $F_\alpha \cap K \neq \emptyset$),

$$(\cap \mathcal{F}) \cap K \neq \emptyset.$$

Definition (Compact open topology)

This is the topology of uniform convergence on linearly compact subspaces.

Bases at zero

- Functionals $\mathcal{L}_c(E, F)$: $W(K, V) = \{f \,|\, f(K) \subset V\}$ with K linear compact and V open subspace.
- Dual space E' : $K^\perp = \{x' \,|\, \forall x \in K,\ x'(x) = 0\}$ with K linearly compact subspace.

Function spaces and Orthogonal

Definition (Linear compactness)

A subspace K of a Lefschetz Space is said *linearly compact* when for every closed affine filter $\mathcal{F} = \{F_\alpha\}$ satisfying the intersection property ($\forall F_\alpha$, $F_\alpha \cap K \neq \emptyset$),

$$(\cap\mathcal{F}) \cap K \neq \emptyset.$$

Definition (Compact open topology)

This is the topology of uniform convergence on linearly compact subspaces.

Bases at zero

- Functionals $\mathcal{L}_c(E, F)$: $W(K, V) = \{f \,|\, f(K) \subset V\}$ with K linear compact and V open subspace.
- Dual space E' : $K^\perp = \{x' \,|\, \forall x \in K,\ x'(x) = 0\}$ with K linearly compact subspace.

Reflexivity problems

Linear Logic model ?

Reflexivity is not ensured in general.

It is preserved by quotient, product.

This model generalizes Finiteness spaces. But we need more constraints to ensure reflexivity.

Conclusion

- From semantics to programming languages and vice versa.
- Application of differential nets (concurrency, ...).
- Work in progress : Interpretation of Polymorphic Lambda-Calculus using Lefschetz Linear Spaces.

Bibliography

- [Girard] *Linear Logic*, Theoretical Computer Science, 1987
- [Ehrhard and Regnier] *The differential λ-calculus*, Theoretical Computer Science, 2003
- [Ehrhard and Regnier] *Differential Interaction Nets*, Electronic Notes in Theoretical Computer Science, 2005
- [Ehrhard] *Finiteness spaces*, Mathematical Structures in Computer Science, 2005