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

1. **Linearity : an analogy**
   - Linearity in Computer Science
   - The Analogy
   - Mathematical Tools

2. **Differential Lambda Calculus**
   - Syntax
   - Reduction
   - Taylor expansion

3. **Differential Proofs Nets**
   - Definition
   - Taylor expansion

4. **Semantics**
   - The seminal semantics : Finiteness Spaces
   - A generalization : Lefschetz Spaces
The Question

How many times a program uses its argument?

Let’s look at an example:

Power:

\[
\begin{align*}
\{ & \mathbb{R} \times \mathbb{N} \rightarrow \mathbb{R} \\
& x, n \mapsto x^n
\end{align*}
\]

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.

\[ [\text{Prog}] : A \Rightarrow B \]

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

\[ [\text{Prog}] : !A \multimap B \]

For instance, Power :

\[ !\mathbb{R} \otimes \mathbb{N} \multimap \mathbb{R} \]
\[ (x, n) \mapsto x^n \]
An Analogy

Mathematical Linearity
A linear function is a first degree polynomial function. Every regular function can be approximated by a linear function:

\[ f(x) \underset{x \to 0}{\approx} f(0) + f'(0) \cdot x \]

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

1. Linearity: an analogy
   - Linearity in Computer Science
   - The Analogy
   - Mathematical Tools

2. Differential Lambda Calculus
   - Syntax
   - Reduction
   - Taylor expansion

3. Differential Proofs Nets
   - Definition
   - Taylor expansion

4. Semantics
   - The seminal semantics: Finiteness Spaces
   - A generalization: Lefschetz Spaces
An extension of $\lambda$-Calculus

Syntax

\[
s, t ::= x | \lambda x.s | (s)t | Ds.t | 0 | as + bt
\]

\[a, b \in R\] where $R$ is a ring.

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 deterministic.
Linear Analogy and Sums

\[ \lambda x. (s + t) = \lambda x.s + \lambda x.t \]  
\[ (s + t)u = (s)u + (t)u \]  
\[ (s)(u + v) \neq (s)u + (s)v \]  

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

\[ D(\lambda x. t).u \rightarrow \lambda x. \left( \frac{\partial t}{\partial x}.u \right) \]  \hspace{0.5cm} (4)

Linear substitution:

The term \( \frac{\partial t}{\partial x}.u \) means one occurrence of \( x \) has been substituted by \( u \) in \( t \). It is a non deterministic operation since there are several occurrences that can be substituted.
Substitutions and Differentiation

Differential reduction

\[ D(\lambda x.t).u \rightarrow \lambda x. \left( \frac{\partial t}{\partial x}.u \right) \]  

(4)

Linear substitution :

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

\[ \frac{\partial y}{\partial x}.u = \delta_{xy}u \]
Substitutions and Differentiation

Differential reduction

\[ D(\lambda x.t).u \rightarrow \lambda x. \left( \frac{\partial t}{\partial x}.u \right) \]  

(4)

Linear substitution :

The term \( \frac{\partial t}{\partial x}.u \) means one occurrence of \( x \) has been substituted by \( u \) in \( t \). It is a non deterministic operation since there are several occurrences 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

\[ D(\lambda x.t).u \rightarrow \lambda x. \left( \frac{\partial t}{\partial x}.u \right) \]  \hspace{1cm} (4)

Linear substitution:

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

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

\[ \rightarrow (f \circ g)'(x) = f'(g(x)) \cdot g'(x) \]
Substitutions and Differentiation

Differential reduction

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

Linear substitution:
The term \( \frac{\partial t}{\partial x}.u \) means one occurrence of \( x \) has been substituted by \( u \) in \( t \). It is a non deterministic operation since there are several occurrences that can be substituted.

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

Differential reduction

\[ D(\lambda x.t).u \rightarrow \lambda x. \left( \frac{\partial t}{\partial x}.u \right) \]  

(4)

Linear substitution:

The term \( \frac{\partial t}{\partial x}.u \) means one occurrence of \( x \) has been substituted by \( u \) in \( t \). It is a non deterministic operation since there are several occurrences 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:

\[(s)u = \sum_{n=0}^{\infty} \frac{1}{n!} (D^n s.u^n)0\]  

(5)

Theorem (Ehrhard, Régnier 2006)

*Purely \(\lambda\)-calculus can be encoded through Taylor Expansion in the purely differential \(\lambda\)-calculus.*
Summary

1. Linearity : an analogy
   - Linearity in Computer Science
   - The Analogy
   - Mathematical Tools

2. Differential Lambda Calculus
   - Syntax
   - Reduction
   - Taylor expansion

3. Differential Proofs Nets
   - Definition
   - Taylor expansion

4. Semantics
   - The seminal semantics : Finiteness Spaces
   - A generalization : Lefschetz Spaces
Linear Logic Nets

A programming language:

\[ A \triangledown B \]
\[ A \otimes B \]
\[ A ? A \]
\[ A ! N B_1 \ldots B_k \]
Differential Nets

A Linearized programming language:

\[ A \boxtimes B \]

\[ A \otimes B \]

\[ A \triangleleft B \]

\[ ?A \]

\[ !A \]

\[ ?B_1 \]

\[ \ldots \]

\[ ?B_k \]
Differential Nets

A Linearized programming language:

\[ A \otimes B \]

\[ A \oplus B \]

\[ A \otimes B \]

\[ A \otimes B \]

\[ A \otimes B \]
The principle:

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

\[ \sum_{n=0}^{\infty} \frac{1}{n!} \]

where $N^*_k$ in Taylor expansion of $N$. 
Differential Nets vs. Differential $\lambda$-Calculus

Theorem (Ehrhard, Régnier 2006)

*Differential $\lambda$-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 $\lambda$-calculus.
- Symmetry between $?$- and $!$-cells that is the monad and the comonad.
- Links with concurrence: $\pi$-calculus can be encoded in differential nets.
Summary

1. Linearity: an analogy
   Linearity in Computer Science
   The Analogy
   Mathematical Tools

2. Differential Lambda Calculus
   Syntax
   Reduction
   Taylor expansion

3. Differential Proofs Nets
   Definition
   Taylor expansion

4. Semantics
   The seminal semantics: Finiteness Spaces
   A generalization: Lefschetz Spaces
History of linear models

Linear Logic

\[
\begin{array}{ccc}
A & |A| & [A] = k^{|A|} \\
\perp & A\perp & |A| \\
\oplus, \& & A \oplus B & |A| + |B| \\
\otimes & A \otimes B & |A| \times |B| \\
\multimap & A \multimap B & |A| \times |B| \\
! & !A & M_f(|A|) \\
\end{array}
\]

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.
Infinite dimension problems

- Which basis notion?
- How to ensure reflexivity?

In order to solve them, we need some topology.


Finiteness Spaces

The relational model viewpoint.

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

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

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

Example: Integers.
The linear spaces view point.
For every \( x \in k^{\lvert X \rvert} \), the support of \( x \) is \( \lvert x \rvert = \{ a \in \lvert X \rvert | x_a \neq 0 \} \).

Definition
The linear space associated to \( X = (\lvert X \rvert, \mathcal{F}(X)) \) is:

\[
k\langle X \rangle = \{ x \in k^{\lvert X \rvert} | \lvert x \rvert \in \mathcal{F}(X) \}.
\]

endowed by the topology generated by the basis at zero:

\[
\{ V_J | J \in \mathcal{F}^\perp \} \text{ where }
\]

\[
V_J = \{ x \in k\langle X \rangle | \lvert x \rvert \cap J = \emptyset \}.
\]

Example: Integers.
A Linear Logic Model

\[ X^\perp \leadsto k\langle X \rangle' \]

\[ 0 \leadsto \{0\} \]

\[ X \& Y \quad X \oplus Y \quad \Rightarrow \quad k\langle X \rangle \oplus k\langle Y \rangle \]

\[ 1 \leadsto k \]

\[ X \multimap Y \quad \Rightarrow \quad \mathcal{L}_c(X, Y) \]

\[ X \otimes Y \quad \Rightarrow \quad k\langle X \rangle \otimes k\langle Y \rangle \]

\[ !X \leadsto k\langle !X \rangle \]

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

where

\[ \mathcal{F}(!X) = \{ A \subseteq \mathcal{M}_{fin}(|X|) \mid \bigcup_{m \in A} |m| \in \mathcal{F}X \} \]
Finiteness Spaces

A Linear Logic Model

\[ X \downarrow \leadsto k\langle X \rangle' \quad \Rightarrow \text{Reflexivity} \]

\[ 0 \leadsto \{0\} \]

\[ X & Y \quad \leadsto \quad k\langle X \rangle \oplus k\langle Y \rangle \]

\[ 1 \leadsto k \]

\[ X \multimap Y \quad \leadsto \quad \mathcal{L}_c(X, Y) \]

\[ X \otimes Y \quad \leadsto \quad k\langle X \rangle \otimes k\langle Y \rangle \]

\[ !X \quad \leadsto \quad k\langle !X \rangle \quad \Rightarrow \text{Infinite dimension} \]

\[ |!X| = \mathcal{M}_{\text{fin}}(|X|) \]

where \[ \mathcal{F}(!X) = \{ A \subseteq \mathcal{M}_{\text{fin}}(|X|) \mid \bigcup_{m \in A} |m| \in \mathcal{F}X \} \]
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.
Linearized topological vector spaces have been introduced by S. Lefschetz in 1942.

They appear in

[Barr] \textit{\textastern-automonomous Categories}, Lecture Notes in Mathematics, 1979


[Ehrhard] \textit{Finiteness spaces}, Mathematical Structures in Computer Science, 2005
Lefschetz spaces

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

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

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

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

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

Example: Finiteness spaces with the basis topology. Finite sequences $k^{(\omega)}$ 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.
Function spaces and Orthogonal

Definition (Linear compactness)

A subspace $K$ of a Lefschetz Space is said \textit{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


