Algebraic λ-calculus
On linear combinations of terms

Lionel Vaux
Institut de Mathématiques de Luminy, Marseille
vaux@iml.univ-mrs.fr

Rewriting Techniques and Applications, 2007, Paris
Background

There are models of linear logic where

- types are interpreted by vector spaces (more generally modules over rigs);
- proofs are interpreted by linear maps;
- linear maps from $!A$ to $B$ are analytic maps from $A$ to $B$.

**Thomas Ehrhard.**

On Köthe sequence spaces and linear logic.

**Thomas Ehrhard.**

Finiteness spaces.
Usual translation of simple types: $A \Rightarrow B = !A \rightarrow B$.

Ehrhard-Regnier’s differential $\lambda$-calculus: terms as analytic functions between vector spaces.
- Differentiation.
- But also sums and linear combinations.

Exists in pure and typed flavours.

Thomas Ehrhard and Laurent Regnier.
The differential lambda-calculus.
Differential calculus

- Usual translation of simple types: $A \Rightarrow B = !A \to B$.
- Ehrhard-Regnier’s differential $\lambda$-calculus: terms as analytic functions between vector spaces.
  - Differentiation.
  - But also sums and linear combinations.
- Exists in pure and typed flavours.
- Reduction may behave strangely...
Differential calculus

- Usual translation of simple types: $A \Rightarrow B = !A \rightarrow B$.
- Ehrhard-Regnier’s differential $\lambda$-calculus: terms as analytic functions between vector spaces.
  - Differentiation.
  - But also sums and linear combinations.
- Exists in pure and typed flavours.
- Reduction may behave strangely... only because of coefficients in linear combinations.

Thomas Ehrhard and Laurent Regnier.
The differential lambda-calculus.
Terms as functions between vector spaces

**Basic ideas**

- Extend the set of terms so that if forms a vector space.
- Mappings with values in a vector space form a vector space:

\[(a.f + b.g)[x] = a.f[x] + b.g[x]\]

In \(\lambda\)-ish words:

\[\lambda x (a.s + b.t) u = a.(s) u + b.(t) u\]
\[\lambda x (a.s + b.t) = a.\lambda x s + b.\lambda x t\]
Definitions

**Definition (Rig)**

We call *rig* any tuple \((R, +, 0, \times, 1)\) where

- \((R, +, 0)\) and \((R, \times, 1)\) are commutative monoids;
- \(0 \times a = 0\) and \((a + b) \times c = (a \times c) + (b \times c)\).

**Definition (Module)**

A module over a rig \(R\) (or \(R\)-module) is a tuple \((V, +, \vec{0}, .)\) where

- \((V, +, \vec{0})\) is a commutative monoid;
- external product \((.)\) is left and right additive: \(0.v = a.\vec{0} = \vec{0}\), \((a + b).v = a.v + b.v\) and \(a.(v + w) = a.v + a.w\);
- \(1.v = v\) and \(a.b.v = (a \times b).v\).
Motivations
Algebraic $\lambda$-calculus
On soundness
On normalization
Other approaches

Algebraic $\lambda$-terms
Extending $\beta$-reduction

Morphology

**Definition (Raw terms)**

Let $R$ be a fixed rig.
The set $\Lambda_R$ of raw terms $(\sigma, \tau, \ldots)$ is given by:

$$\sigma, \tau ::= x \mid \lambda x \sigma \mid (\sigma) \tau \mid \vec{0} \mid a.\sigma \mid \sigma + \tau .$$

We will often write $\sum_{i=1}^{n} a_i.\sigma_i$ for

$$a_1.\sigma_1 + \cdots + a_n.\sigma_n + \vec{0} .$$

**Definition (Structural equality)**

We write $\sim$ for the transitive closure of $\alpha$-equivalence and AC of $+$. 
Definition

We define sets $A_R$ of atomic terms $(s, t, \ldots)$ and $C_R$ of of canonical terms $(S, T, \ldots)$ by:

- any variable $x$ is an atomic term;
- if $x \in V$ and $s \in A_R$ then $\lambda x \ s \in A_R$;
- if $s \in A_R$ and $T \in C_R$ then $(s) \ T \in A_R$;
- if $a_1, \ldots, a_n \in R^*$ and $s_1, \ldots, s_n \in A_R$ are pairwise distinct ($\not\sim$) then $\sum_{i=1}^{n} a_i \cdot s_i \in C_R$.

If $s \in A_R$ we write $\vec{s} = 1.s + \vec{0} \in C_R$. 
Canonization

Define $\text{can} : \Lambda_R \rightarrow C_R$ inductively

- $\text{can}(x) = \vec{x}$;
- if $\text{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ then $\text{can}(\lambda x \sigma) = \sum_{i=1}^{n} a_i.\lambda x s_i$;
- if $\text{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ and $\text{can}(\tau) = T$
  then $\text{can}((\sigma)\tau) = \sum_{i=1}^{n} a_i.(s_i) T$;
- $\text{can}(\vec{0}) = \vec{0}$;
- if $\text{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ and $\text{can}(\tau) = \sum_{i=n+1}^{n+p} a_i.s_i$ then

$$\text{can}(\sigma + \tau) = \text{cansum} \left( \sum_{i=1}^{n+p} a_i.s_i \right) ;$$

- if $\text{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ then $\text{can}(a.\sigma) = \text{cansum} (\sum_{i=1}^{n} (a \times a_i).s_i)$. 
Implementation of algebraic identities

Representation

We write $\sigma \equiv_R \tau$ iff $\text{can}(\sigma) \sim \text{can}(\tau)$. Then:

- $\equiv_R$ is an equivalence relation;
- $(\Lambda_R/\equiv_R, +, 0, \cdot)$ is an $R$-module;
- each $\equiv_R$-class has a unique canonical representative (up to $\sim$).

Algebraic terms

We now consider terms up to $\equiv_R$:

- we write $\Delta_R$ for the set of all $\equiv_R$-classes of atomic terms (which we call simple terms);
- $\Lambda_R/\equiv_R$ is generated by $\Delta_R$;
- we call algebraic terms the elements of $R\langle \Delta_R \rangle = (\Lambda_R/\equiv_R)$. 

Lionel Vaux (IML)
Idea

Define a reduction relation on $R\langle \Delta_R \rangle$ such that:

- if $\sigma$ is simple then $(\lambda x \sigma) \tau \rightarrow \sigma[x := \tau]$;
- if $\sigma$ is simple, $\sigma \rightarrow \sigma'$ and $a \neq 0$ then $a.\sigma + \tau \rightarrow a.\sigma' + \tau$.

Warning

It cannot be defined by induction on terms: if $a, b \in R^\bullet$ with $a + b = 0$ then $\vec{0} =_R a.\sigma + b.\sigma$ may reduce.
Extending $\beta$-reduction

Definition

We define $\rightarrow$ on algebraic terms by the following statements:

Reduction of simple terms:

- $$(\lambda x \ s) \ T \rightarrow s \ [x := T];$$
- if $s \rightarrow S'$ then
  
  $$\lambda x \ s \rightarrow \lambda x \ S'$$
  $$s \ T \rightarrow (S') \ T$$
  $$a \cdot s + T \rightarrow a \cdot S' + T \text{ as soon as } a \neq 0$$

Extension to all terms: if $T \rightarrow T'$ then $(s) \ T \rightarrow (s) \ T'$. 
Examples

- Every ordinary $\beta$-reduction is a valid reduction of algebraic $\lambda$-calculus: if $s \rightarrow_{\Lambda} t$ then $s \rightarrow t$.
- If $R = \mathbb{Z}$, and $s \rightarrow S'$:
  \[
  \bar{0} =_{R} s - s \rightarrow S' - s.
  \]
- More generally, if $\exists a \in R \text{ s.t. } a + 1 = 0$:
  \[
  S' =_{R} (s + as) + S' \rightarrow s + (aS' + S') =_{R} s.
  \]
- If $R = \mathbb{Q}$ and $s \rightarrow S'$:
  \[
  s =_{R} \frac{1}{2} s + \frac{1}{2} s \rightarrow \frac{1}{2} s + \frac{1}{2} S' \rightarrow \frac{1}{4} s + \frac{3}{4} S' \rightarrow \cdots
  \]
Confluence

Tait–Martin-Löf

Introduce parallel reduction $\Rightarrow$ such that

$$\rightarrow \subset \Rightarrow \subset \rightarrow^*.$$  

Denote by $S \downarrow$ the term obtained by firing all redexes in $S$.

Lemma

For all terms $S$ and $S'$ such that $S \Rightarrow S'$, we have $S' \Rightarrow S \downarrow$.

This holds only thanks to the way we reduce linear combinations.

Theorem

Reduction $\rightarrow$ enjoys Church-Rosser.
Positive rig

Definition

A rig \( R \) is said to be positive if \( a + b = 0 \) implies \( a = b = 0 \).

Examples:

- The set \( \mathbb{N} \) of natural integers.
- Sets \( \mathbb{Q}^+ \) and \( \mathbb{R}^+ \) of non-negative numbers.
- The set \( R[\xi_0, \xi_1, \ldots] \) of polynomials over indeterminates \( \xi_0, \xi_1, \ldots \), with coefficients taken in a positive rig \( R \).
Conservativity

Assume R is positive.

Lemma

If \( s \in \Lambda \) and \( s \rightarrow^* S' \) then there is \( t \in \Lambda \) such that \( S' \rightarrow^* t \) and \( s \rightarrow^* \Lambda t \).

Theorem

If \( s, s' \in \Lambda \), then \( s \leftrightarrow s' \) iff \( s \leftrightarrow\Lambda s' \).

Corollary

Reductional equality is sound.
Indeterminate forms

Something is rotten in the state of Denmark.

- Let $Y$ be a fixed point of ordinary $\lambda$-calculus. Write $\infty_\sigma = (Y)(\lambda x (x + \sigma))$:

$$\infty_\sigma \rightarrow \sigma + \infty_\sigma.$$

- Assume $-1 \in \mathbb{R}$:

$$0 =_{\mathbb{R}} \infty_\sigma - \infty_\sigma!$$
Indeterminate forms

Something is rotten in the state of Denmark.

- Let $Y$ be a fixed point of ordinary $\lambda$-calculus. Write $\infty_\sigma = (Y)(\lambda x \ (x + \sigma))$:
  \[
  \infty_\sigma \rightarrow \sigma + \infty_\sigma.
  \]

- Assume $-1 \in R$:
  \[
  0 =_R \infty_\sigma - \infty_\sigma!\]

Theorem

Assume $R$ is not positive and $a, b \in R^\bullet$ are such that $a + b = 0$. Then, for all $\sigma \in R\langle \Delta_R \rangle$, $\vec{0} \rightarrow^* a.\sigma$ and $a.\sigma \rightarrow^* \vec{0}$. 

Lionel Vaux (IML)
Motivations
Algebraic λ-calculus
On soundness
On normalization
Other approaches

Conditions for strong normalization
A weak normalization scheme

Typing

\[ \Gamma, x : A \vdash x : A \]
\[ \Gamma \vdash \lambda x \sigma : A \Rightarrow B \]

\[ \Gamma \vdash \sigma : A \Rightarrow B \]
\[ \Gamma \vdash \tau : A \]
\[ \Gamma \vdash (\sigma) \tau : B \]

\[ \Gamma \vdash 0 : A \]
\[ \Gamma \vdash \sigma : A \]
\[ \Gamma \vdash a.\sigma : A \]

\[ \Gamma \vdash \sigma : A \]
\[ \Gamma \vdash \tau : A \]
\[ \Gamma \vdash \sigma + \tau : A \]
Necessary conditions

Positivity

If $R$ is not positive, every term reduces. Moreover, typability isn’t compatible with $\equiv_R$.

Finite splitting

If $R = \mathbb{Q}^+$ and $s \rightarrow S'$:

$$s =_R \frac{1}{2} s + \frac{1}{2} s \rightarrow \frac{1}{2} s + \frac{1}{2} s' \rightarrow \frac{1}{4} s + \frac{3}{4} s' \rightarrow \ldots$$

Hence, for all $a \in R$,

$$\{(a_1, \ldots, a_n) \in (R^\bullet)^n ; \ n \in \mathbb{N} \text{ and } a = a_1 + \cdots + a_n\}$$

must be finite.
Sufficient conditions

Definition (Width)
Define $w : \mathbb{R} \rightarrow \mathbb{N}$ by:

$$w(a) = \max \{ n \in \mathbb{N} ; \exists (a_1, \ldots, a_n) \in (\mathbb{R}^\cdot)^n ; a = a_1 + \cdots + a_n \}.$$ 

Theorem
If $w$ is a morphism of rigs, then all typable terms are SN.

Example
$\mathbb{N}[\xi_1, \xi_2, \ldots]$. 
Sketch of proof

Adapt your own favourite proof by reducibly, using the following lemma.

Lemma

Write $N_R$ for the set of simple SN terms. Then the set of all SN terms is $R\langle N_R \rangle$.

Proof: Let $S \in R\langle N_R \rangle$. One proves that $S$ is SN by induction on

$$\sum_{s \in A_R} w(S_s) |s| .$$
Assume $R$ is positive and $\sigma \in R\langle \Delta_R \rangle$ is typable.

Algorithm

- Replace scalars occurring in $\text{can}(\sigma)$ with formal pairwise distinct indeterminates $(\xi_1, \xi_2, \ldots)$.
- The object $\tau$ thus obtained lies in $R'\langle \Delta_{R'} \rangle$, where $R' = \mathbb{N}[\xi_1, \xi_2, \ldots]$.
- $\tau$ is also typable and SN applies in $R'\langle \Delta_{R'} \rangle$.
- Replace indeterminates by their values in the NF of $\tau$: this is the normal form of $\sigma$. 
Assume:
- $(+, \vec{0}, .)$ are part of the syntax and make the set of terms a $R$-module;
- reduction is contextual;
- ordinary $\beta$-reductions are valid reductions.

Then, as soon as $-1 \in R$:

$$(\lambda x \ x) \infty y + (-1) \infty y \rightarrow^* \begin{cases} \vec{0} \\ y \end{cases}$$

Hence the calculus is either unsound or non confluent.
Failure is expected!

Assume:
- $(+, \vec{0}, .)$ are part of the syntax and make the set of terms a $R$-module;
- reduction is contextual;
- ordinary $\beta$-reductions are valid reductions.

Then, as soon as $-1 \in R$:

$$(\lambda x x) \infty y + (-1) \infty y \rightarrow^* \{ \vec{0} \} y$$

Hence the calculus is either unsound or non confluent.

Something has to fail.
## Motivations
- Algebraic $\lambda$-calculus
- On soundness
- On normalization
- Other approaches

## Drop confluence

**Motto:** Reduce only in canonical forms.

**Details**
- Consider only atomic and canonical terms.
- Define reduction in the natural way (canonize after each elementary reduction step).

**Expected outcome**
- Typable terms are SN.
- Conservative over ordinary $\lambda$-calculus.
- Can be somehow simulated in our setting as a *reduction strategy* (cf. weak normalization).
- Not confluent (whatever $R$).
Motto: Avoid fixed points by typing.

Details
Consider Church-style terms up to typed algebraic equality: types are R-modules.

Expected outcome
- Should be SN, hence sound.
- Confluence?
Drop contextuality

Motto: Perform algebraic rewriting only on values.

Details

Adapt Arrighi-Dowek’s work on linear algebraic \( \lambda \)-calculus.

- Work on raw terms (equality: \( \sim \)).
- Introduce algebraic equality as part of reduction, allowing algebraic rewriting steps only on closed normal terms.

Pablo Arrighi and Gilles Dowek.
Linear-algebraic lambda-calculus: higher-order, encodings and confluence.
Manuscript, 2006.
Expected outcome

- Conservative over ordinary $\lambda$-calculus.
- Confluent and sound.
- The set of terms is not an $R$-module.
- Reduction is not contextual.
- Normalization properties?
Motto : $\beta$-reduction upto algebraic equality.

Details
This was the topic of the talk.

Outcome
- The set of terms is R-module w.r.t. syntactic sum and external product.
- Contextual and confluent reduction.
- Sometimes not conservative over ordinary $\lambda$-calculus, and even unsound.

THE END